PC-8801

ディスク N88-BASIC 解析マニュアル

山内 直 著

SHUWA
SYSTEM
TRADING
CO.,LTD.

ISBN4-87966-008-6 C0000 ¥3200E

PC-8801

N88-BASIC DISK ANALYTICAL MANUAL

SHUWA SYSTEM TRADING CO.,LTD.

まえがき

１９８１年１２月に，日本電気のベストセラー機ＰＣ－８００１の上位機種として発表されて以来，その多機能性と汎用性の高さで広範囲に渡るユーザ層の支持を受け，愛用され続けてきた日本電気のパーソナルコンピュータ・ＰＣ－８８０１も，発表後すでに２年の年月が過ぎ去ろうとしています．さらに，その後の１６ビットＣＰＵを搭載したＰＣ－９８００シリーズ，Ｎ５２００モデル０５の発表にもかかわらず，究極の８ビット機として，またＰＣシリーズの中心機種として，変わらずの人気を保持しています．

ＰＣ－８８０１には，ミニフロッピディスク用のインタフェースが標準で装備され，ＰＣ－８０Ｓ３１などのミニフロッピディスクユニットを直接接続することができます．また，ＰＣ－８８８１などの標準フロッピディスクユニットもインタフェースの装着により接続することができ，これでＰＣ－８８０１には４種類のディスクドライブが接続可能となりました．

これらのフロッピディスク装置は，最近コストの低下により急激に普及率が高まってきており，カセットテープにかわる手軽な入出力装置としての役割が大きくなってきています．それだけに，ディスクを使いこなしたい，できればマシン語でと考えているユーザが増えてきているのも当然のことと言えるでしょう．

しかし，フロッピディスク装置はその高性能がゆえにコントロールが厄介で，マシン語によるプログラミングの壁となっていることも事実です．

本書は，ディスクを使うにあたっての手軽な入門書としてはもちろん，ディスクを高度に使いこなしたいと考えているユーザの方々にまで対応できるよう，初歩的な事柄からN_{88}－ＤＩＳＫ　ＢＡＳＩＣインタプリタの解析資料に至るまで，実際の使用例と共にここに発表するものです．

本書が読者諸兄のもとで，ＰＣ－８８０１ディスクシステムのより高度な応用への一助となれば幸いです．

最後になりましたが，本書の執筆出版にあたりお世話になりました石井晴正氏，橋本正樹氏，水上英樹氏には，厚く御礼申し上げます．

１９８３年１０月　　　　著者　山内　直

本書の執筆にあたり，使用したシステムは以下に示すものです．

ＰＣ－８８０１	本体（Serial NO.Ｋ２０６３１８２）
ＰＣ－８８０１－０１	漢字ＲＯＭボード
ＰＣ－８８２２	１８ピン・ドットマトリクス漢字プリンタ
ＰＣ－８８５１	１４インチ・モノクロ専用高解像度ディスプレイ
ＰＣ－８８８１	８インチ標準フロッピディスクユニット
ＰＣ－８０Ｓ３１	画面倍密度デュアルミニディスクユニット
ＰＣ－８８８４	８インチシステムディスク
ＰＣ－８８３４－２Ｗ	ＰＣ－８０Ｓ３１用システムディスク

システムディスクは［Aug. ２０，１９８２］Versionを使用し，システムの書き換えなどは一切行われていないものを使用しました．

また，以下の文献を引用もしくは参考とさせて頂きました．

ＰＣ－８８０１　ＢＡＳＩＣリファレンスマニュアル　日本電気（株）

ＰＣ－８８０１　ユーザーズマニュアル　日本電気（株）

トランジスタ技術　’８３　１月号　ＣＱ出版社

ＰＣ－８８０１　Ｎ88－ＢＡＳＩＣ解析マニュアル

ＰＣ－９８０１　解析マニュアル［第０巻］

ＰＣ－８００１　マシン語活用ハンドブック（初級編）内部サブルーチンのすべて．

ＰＣ－８００１　マシン語活用ハンドブック（中級編）インタフェースの実際．

以上４著　秀和システムトレーディング（株）

— 目　次 —

— Contents —

第３章　N88－DISK BASICインタプリタの解析……45

— Analysis of N88－DISK BASIC Interpreter —

第6章　付録……………………………………………………269

— Appendix —

SAMPLE PROGRAMS

第1章　N88-Disk BASICのファイル構造

——File Structure of N88-Disk BASIC——

1－1　ディスクの管理状況.

N_{88}－DISK BASICでは，1枚のディスケットを次の様に区分し，1セクタを基本単位としてディスクを管理しています．また，N_{88}－DISK BASICでは，5インチのDMAタイプドライブについても考慮されていますが，その管理については5インチのインテリジェントタイプドライブ（但し両面）のものと同一と見るのが適当です．以下に，システムディスクの場合のディスク・マップを各ドライブタイプ毎に示します．

〔8インチ・両面倍密度〕

サーフェス	クラスタ	トラック	セクタ	用途
0	0	0	1～26	末使用
	2	1	1	IPL
			2～26	ディスクコード
	4	2	1～26	
	6～68	3～34	1～26	ユーザ領域
	70	35	1～22	ディレクトリ
			23	ID
			24～26	FAT
	72～152	36～76	1～26	ユーザ領域
1	1	0	1～26	未使用
	3	1	1～26	ディスクコード
	5～153	2～76	1～26	ユーザ領域

Fig. 1－1－1　8″両面倍密度

ここで，トラック0はサーフェス0，1共に未使用となっていますが，これはディスクのフォーマットが他の部分と異なるためです．特にサーフェス0に関しては，1セクタのデータ長が128バイト（単密）となっているため，通常の方法では読み書きすることは不可能です（ただし，FDCを直接制御する場合は別となりますが，BASICからは使用されていませんので特別に考慮する必要はありません）．

〔5インチ・両面倍密度〕

サーフェス	クラスタ	トラック	セクタ	用　　途
0	0	0	1～16	IPL
	1		2～16	ディスクコード
	4～5	1	1～16	
	8～157	2～39	1～16	ユーザ領域
1	2～3	0	1～16	ディスクコード
	6～7	1	1～16	
	10～71	2～17	1～12	ユーザ領域
	74	18	1	ディレクトリ
	75		13	ID
			14～16	FAT
	78～159	19～39	1～16	ユーザ領域

Fig. 1－1－2　5″両面倍密度

〔5インチ・片面倍密度〕

クラスタ	トラック	セクタ	用　　途
0	0	1	IPL
1		2～16	ディスクコード
2～3	1	1～16	
4～5	2	1～16	
6～7	3	1～16	
8～35	4～17	1～16	ユーザ領域
36	18	1～12	ディレクトリ
37		13	ID
		14～16	FAT
38～69	19～34	1～16	ユーザ領域

Fig. 1－1－3　5″片面倍密度

※使用するディスクがデータディスクの場合には，ＩＰＬ・ディスクコード領域も全てユーザ領域となります．

1－2　トラック・セクタ.

N$_{88}$－DISK BASICでディスクを扱う際，その最小単位は１セクタとなっています．ディスクが倍密度の場合，１セクタあたりのデータ長は256バイトとなっているので，256バイト単位（１００H）でデータの入出力が行えます．

そして，１トラックには８インチの場合で２６，５インチの場合で１６のセクタが存在し，８インチ両面の場合で０（１）～７７の７７（７８）トラック，５インチ両面の場合で０～３９の４０トラック，５インチ片面の場合で０～３４の３５トラックが存在しています．

また，両面仕様においては，それぞれのトラックにサーフェス０とサーフェス１という２つのトラックが存在し，それぞれ別のトラックとして扱うことができます．しかし，両面仕様の場合においても片面仕様の場合と同様にトラックの指定が行えるように，トラック番号にサーフェス番号の意味を含んだ単位を設定しています．それが論理トラック番号で，それらは次の様に表せます．

〈論理トラック番号〉＝〈物理トラック番号〉×２
＋〈サーフェス番号〉

ここでは，論理トラック番号に対し，通常のトラック番号を物理トラック番号もしくはシリンダ番号と呼ぶことにします．マシン語レベルでディスクを扱う場合，トラックの指定は全て論理トラック番号で指定します．それにより，サーフェス０とサーフェス１を交互にアクセスするため，ヘッドの移動が少なくて済むという利点があります．

最後に，論理トラック番号のビット構成を示しておきます．

論理トラック番号
（０～１５９，１５３）

才１～才７ビット
シリンダ番号
（０～７９，７７）

才０ビット
サーフェス番号
（０～１）

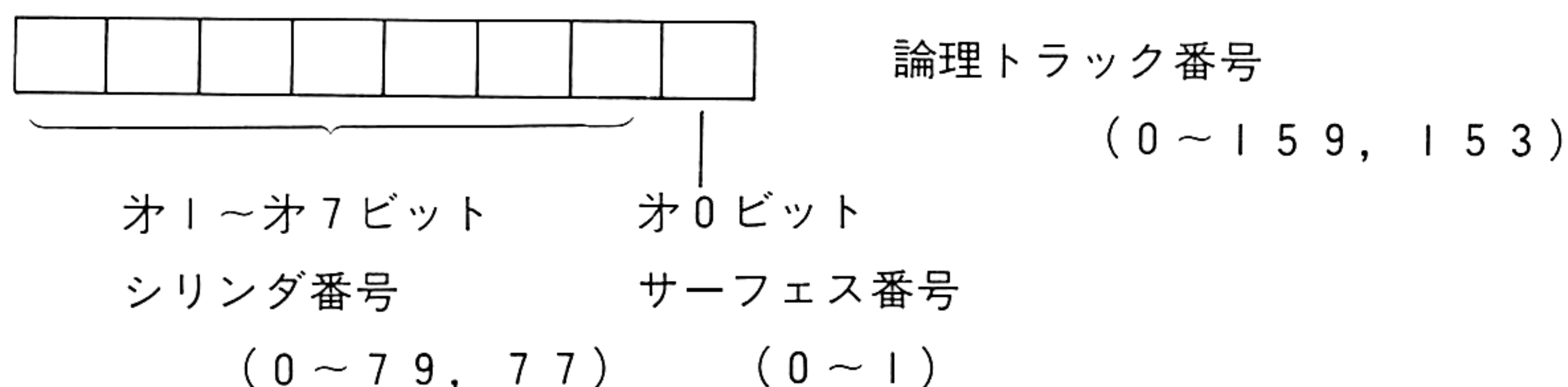



Fig. １－２－１　論理トラック番号

1－3　クラスタ．

N₈₈－DISK BASICでディスクを扱う際の最小単位は1セクタであると述べましたが，これら1つ1つを管理すると，ディスケットの使用効率が高くなるという反面，8インチにおいては4000近くものセクタを管理することになり，相対的にディスクアクセスの速度が低下したりディスク上のこれらの管理表の占める割合も軽視できなくなるという点も出てきます．

そこで，N₈₈－DISK BASICではファイルを扱う場合にのみ，クラスタという別の単位を設けています．この大きさはドライブのタイプにより異なり，8インチの場合で26セクタ，5インチの場合で8セクタとなっています．また，8インチの場合は，

〈クラスタ番号〉＝〈物理トラック番号〉×2
＋〈サーフェス番号〉

と表せますが，5インチの場合は少々複雑となります．以下に，各ドライブタイプのクラスタ番号のビット構成を示します．

〔8インチ両面〕

クラスタ番号（0～153）

オ1～オ7ビット

トラック番号（0～77）

第0ビット

サーフェス番号（0～1）

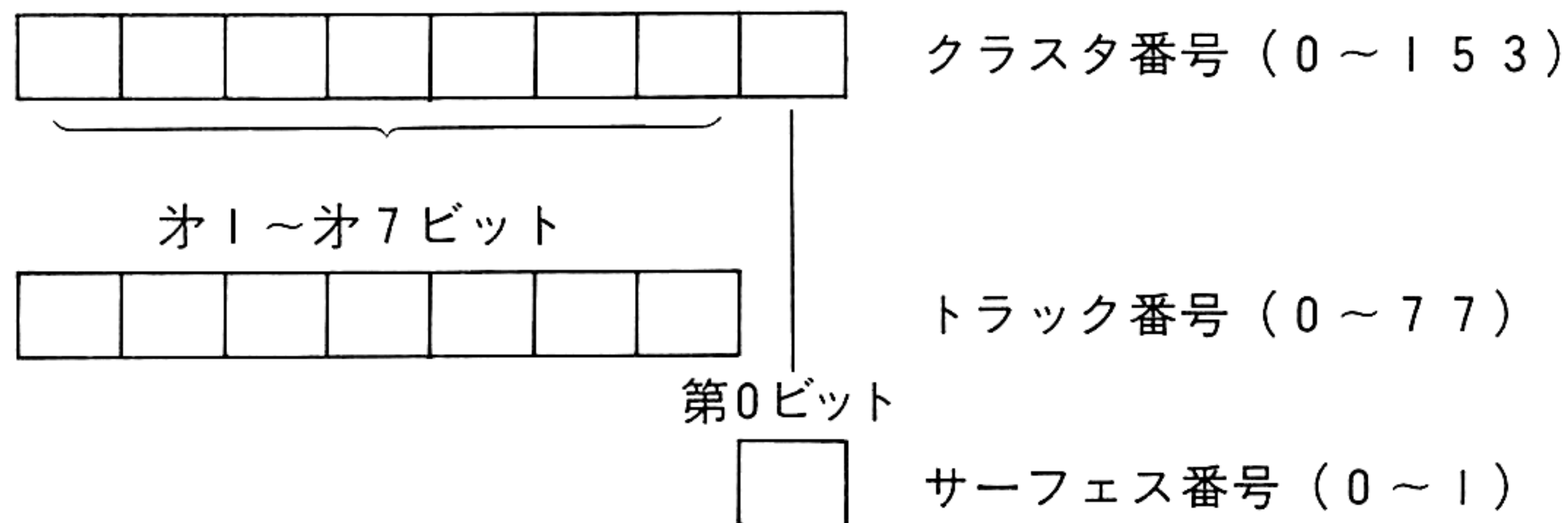



Fig. 1－3－1　8″両面ディスクのクラスタ構成

〔5インチ両面〕

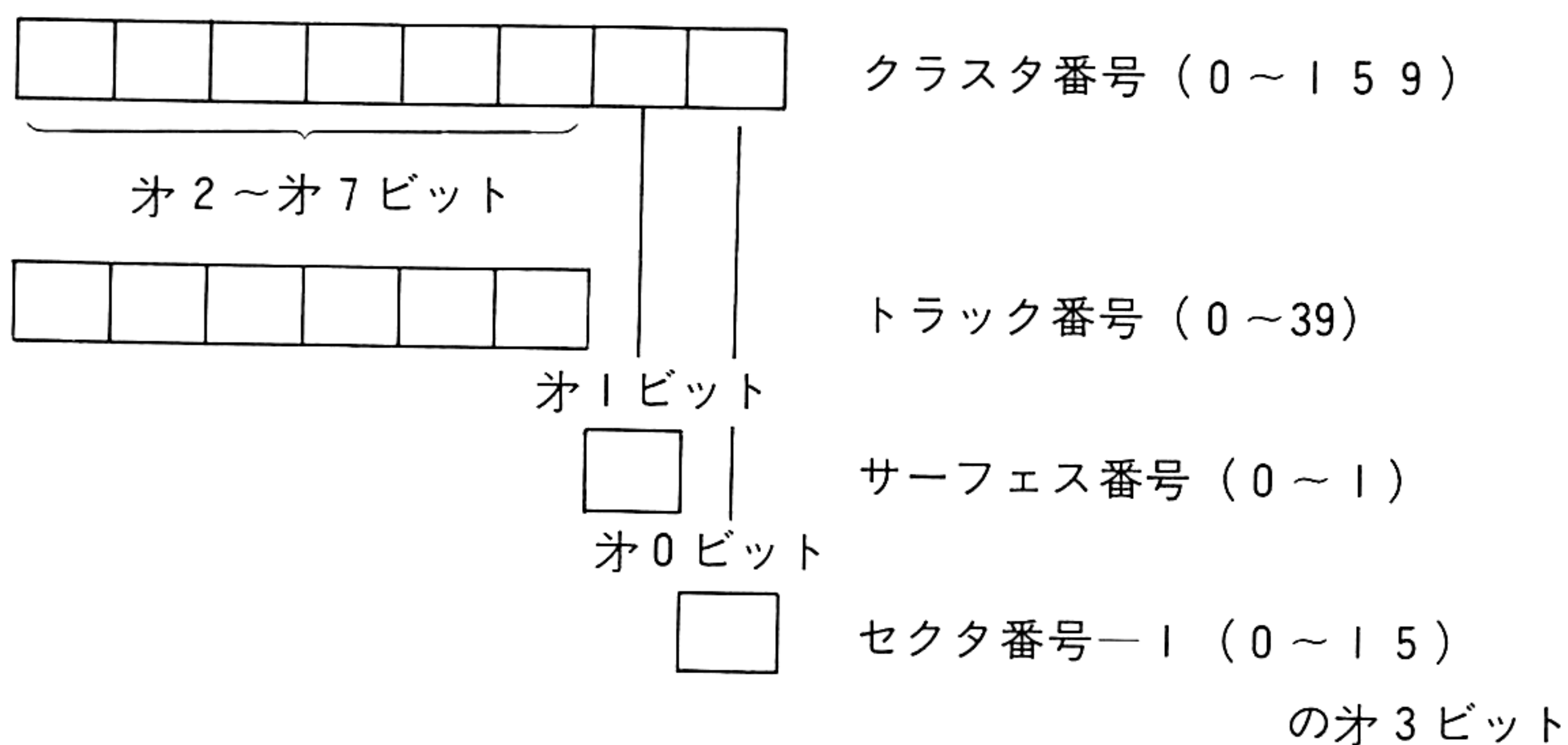



Fig. 1－3－2　5″両面ディスクのクラスタ構成

〔5インチ片面〕

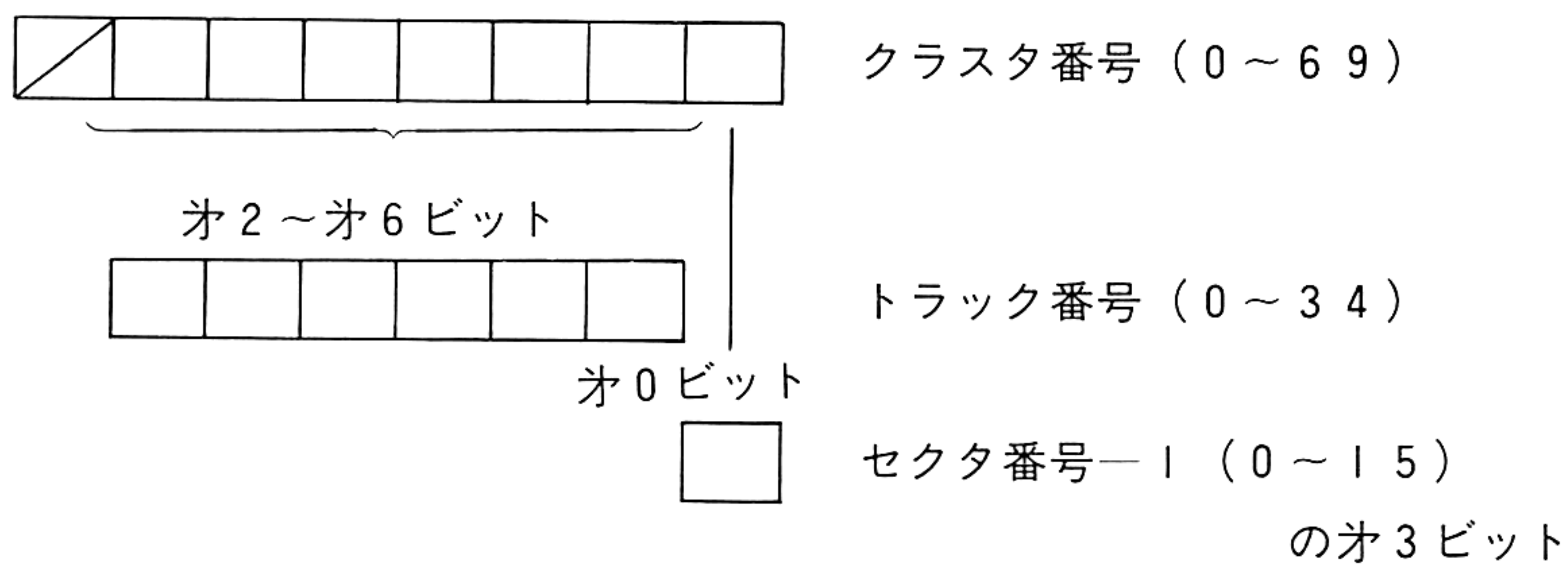



Fig. 1－3－3　5″片面ディスクのクラスタ構成

1－4　ディレクトリ.

ディレクトリというのは，そのディスクに格納されているファイルの名前，形式，属性，格納されている先頭クラスタ番号を記録してある表で，ドライブのタイプに応じて次の様に割り当てられています.

8インチ両面……サーフェス0，トラック35，セクタ1～22

5インチ両面……サーフェス1，トラック18，セクタ1～12

5インチ片面……トラック18，セクタ1～12

また，1つのファイルには16バイトずつ割り当てられており，1セクタのディレクトリで16個のファイルを管理することができます．次に，16バイト中の各バイトの意味を示します.

ファイル名
格張子
属性
先頭クラスタ番号
最初1バイトが，
00H………KILLされたファイル
FFH………未使用
それ以外………有効なファイル
マシン語ファイル
P属性
プロテクトファイル
R属性
非アスキー形式ファイル

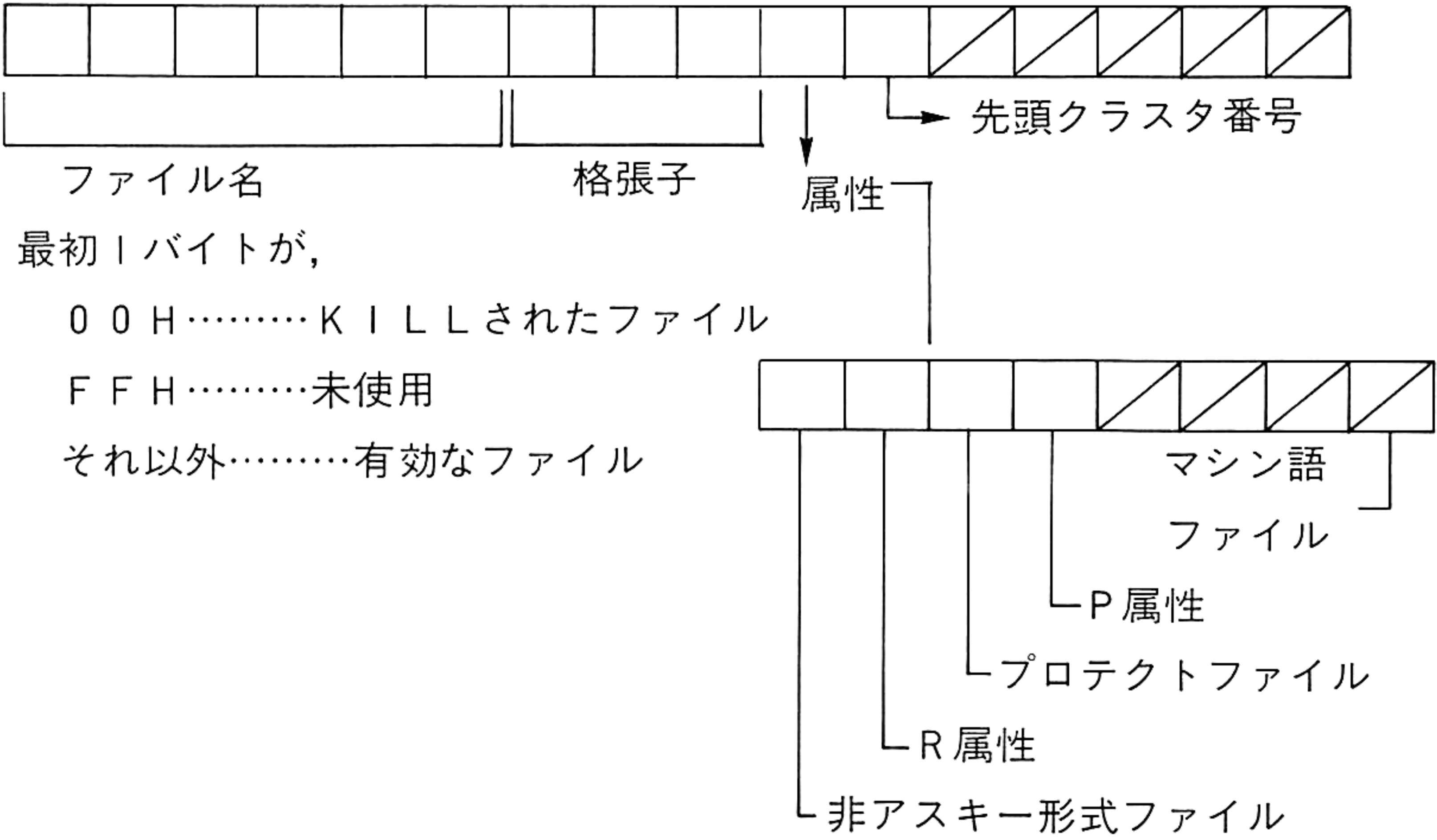



Fig. 1－4　ディレクトリの構造

ディレクトリの最初の1バイトがFFHであった場合，それはそのディレクトリが未使用であることを示すばかりでなく，以下のディレクトリも未使用であることを示します.

では，実際にディレクトリがどの様に使用されているかを見てみることにします．これは，ファイルが次の様に格納されているディスクのディレクトリです(ただし5インチ両面).

```
backup.n88 2     format.n88 2     setinf.n88 1     xfiles.n88 1     Quartz.     1
Cpaint.     1    DEMOUT.     6    sysgen.     1    CNVDSK.n88 1     mnemon ic   3
exdcom.bas 1     dskedt.bas 1     dskedt*bin 1

Track 12,Surface 01,Sector 01
0000 62 61 63 6B 75 70 6E 38 38 80 48 FF FF FF FF FF        backupn88_H
0010 66 6F 72 6D 61 74 6E 38 38 80 49 FF FF FF FF FF        formatn88_I
0020 73 65 74 69 6E 66 6E 38 38 80 45 FF FF FF FF FF        setinfn88_E
0030 78 66 69 6C 65 73 6E 38 38 80 44 FF FF FF FF FF        xfilesn88_D
0040 51 75 61 72 74 7A 20 20 20 80 4C FF FF FF FF FF        Quartz    _L
0050 43 70 61 69 6E 74 20 20 20 80 43 FF FF FF FF FF        Cpaint    _C
0060 00 45 4D 4F 20 20 20 20 20 80 4D FF FF FF FF FF         EMO      _M
0070 44 45 4D 4F 55 54 20 20 20 80 42 FF FF FF FF FF        DEMOUT    _B
0080 73 79 73 67 65 6E 20 20 20 80 41 FF FF FF FF FF        sysgen    _A
0090 43 4E 56 44 53 4B 6E 38 38 A0 4D FF FF FF FF FF        CNVDSKn88 M
00A0 6D 6E 65 6D 6F 6E 69 63 20 10 4E FF FF FF FF FF        mnemonic  N
00B0 65 78 64 63 6F 6D 62 61 73 80 3E FF FF FF FF FF        exdcombas_>
00C0 00 78 64 63 6F 6D 62 69 6E 01 3D FF FF FF FF FF         xdcombin =
00D0 64 73 6B 65 64 74 62 61 73 D0 53 FF FF FF FF FF        dskedtbasミS
00E0 64 73 6B 65 64 74 62 69 6E 01 54 FF FF FF FF FF        dskedtbin T
00F0 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
```

７番目のファイルを見てみますと，その最初のバイトが００Ｈとなっています．これはＫＩＬＬされたファイルを意味します．また，１６番目からは１６バイト全部がＦＦＨとなっており，この部分以降は未使用だということになります．

各ファイルの１０バイト目に着目すると，１～１０番目のファイル，１２番目，１４番目のファイルはバイナリ形式（非アスキー形式）であることがわかり，１０番目に関しては，さらにＰオプション付でセーブされていることがわかります．また，１１番目はアスキー形式で，Ｒ属性が指定されていることがわかります．１５番目はマシン語ファイルです．このように，１０バイト目は様々な意味を持っているので，その構成が少々複雑となります．次に，主な値の意味する属性，ファイル形式を示します．

８０Ｈ………非アスキー形式
Ａ０Ｈ………Ｐオプション付き非アスキー形式
Ｃ０Ｈ………Ｒ属性指定非アスキー形式
９０Ｈ………Ｐ属性指定非アスキー形式
０１Ｈ………マシン語ファイル
００Ｈ………アスキー形式

1－5　ＦＡＴ（File allocation table）．

現在そのディスクにおいて，どのクラスタが使用され，またどのようにつながっているかを記録した表がＦＡＴです．ＦＡＴは，そのディスクのタイプに応じて次の様に割り当てられており，３セクタは全て同じ内容です．

８インチ両面………サーフェス０，トラック３５，セクタ２４～２６
５インチ両面………サーフェス１，トラック１８，セクタ１４～１６
５インチ片面………トラック１８，セクタ１４～１６

ＦＡＴは，そのディスクの最大クラスタ番号＋１バイト分の大きさを持ち，その各１バイト毎が１クラスタに対応しています．また，各バイトはそのクラスタの使用・未使用の区別や，複数のクラスタを使用しているファイルでは，後に続くクラスタの番号や，最後のクラスタで実際に使用されているセクタの数を示しています．各バイトの値によるクラスタの情報は次の様になります．

値	意味
８インチ両面００Ｈ～９９Ｈ ５インチ両面００Ｈ～９ＦＨ ５インチ片面００Ｈ～４５Ｈ	そのクラスタがファイルの一部として使用中であることを示します．また，これが連続したクラスタの一部であり，その値が後続するクラスタの番号を示しています．
８インチ両面Ｃ１Ｈ～ＤＡＨ ５インチ　　Ｃ１Ｈ～Ｃ８Ｈ	そのクラスタがファイルの一部として使用中であることを示します．上のものと異なる点は，そのクラスタが連続したクラスタの最後のクラスタを示しているところです．また下位５ビット（５インチの場合は４ビット）が，そのクラスタ内で実際に使用されているセクタ数を示します．
〔８・５インチＦＥＨ〕	予約済のクラスタで，これらはファイルとして使用することはできません．ＩＰＬ・ディスクコード・ディレクトリ・ＩＤ・ＦＡＴを含むクラスタがこれに当たります．
〔８・５インチＦＦＨ〕	そのクラスタが未使用であることを示します．

ここでは,「連続したクラスタ」という表現を使用しましたが,必ずしも物理的に隣り合ったクラスタを意味しません．ふつうN_{88}－DISK BASICのファイルは物理的に断続的（勿論隣り合っている場合もあります）なクラスタをもって構成されています．

では，実際にＦＡＴがどのように使用されているかを見てみることにします．使用したディスクはディレクトリの項で使用したものと同一で，５インチ両面のものです．ここでは，〝ＤＥＭＯＵＴ〞というファイルに着目してみることにします．

```
0070 44 45 4D 4F 55 54 20 20 20 80 42 FF FF FF FF FF        DEMOUT    _B


Track 12,Surface 01,Sector 0E
0000 FE FE FE FE FE FE FE FE FF FF FF FF FF FF FF FF
0010 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
0020 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
0030 FF FF FF FF FF FF FF FF C7 FF FF 38 3B FF C3 3C             ヌ  8; テく
0040 3F C4 40 C3 C8 C5 C7 C6 47 46 FE FE C7 C4 4F 50        ?ト@テネナヌニGF   ヌトOP
0050 C2 FF FF C2 C2 FF FF FF FF FF FF FF FF FF FF FF        ツ  ツツ
0060 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
0070 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
0080 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
0090 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
00A0 00 00 00 FF 00 81 FE FE FE FE FE FE FE FE FF FF              -
00B0 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
00C0 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
00D0 FF FF FF FF FF FF FF FF FF FF FF FF FF FF C7 FF                     ヌ
00E0 FF 38 3B FF C3 3C 3F C4 40 C3 C8 C5 C7 C6 47 46         8; テく?ト@テネナヌニGF
00F0 FE FE C7 FF FF FF C4 52 C8 C2 C2 FF FF FF FF FF          ヌ   トRネツツ
```

ディレクトリの１１バイト目を見ると，そこには４２Ｈという値が書き込まれています．すなわち，このファイルはクラスタ４２Ｈから格納されていることになります．そこで，ＦＡＴの４２Ｈバイト目を見ると４０Ｈ，４０Ｈバイトを見ると３ＦＨ，という具合に

４２Ｈ→４０Ｈ→３ＦＨ→３ＣＨ→３ＢＨ→３８Ｈ

と続き，また，３８ＨにはＣ７Ｈが書き込まれていることから，最後のクラスタであることを示しています．すなわち，〝ＤＥＭＯＵＴ〞というファイルは６クラスタを占め，最後のクラスタでは，このディスクは５インチであるため７セクタを使用しているということになります．

この場合，Ａ０Ｈ以降にも何か書かれていますが，これらは全く意味を持ちません．

1－6　IDセクタ.

ＩＤセクタには１セクタが割り当てられており，そのディスクの属性，N_{88}－DISK BASICスタート時の設定ファイル数・実行テキストが格納されています．ＩＤセクタはディスクのタイプに応じて次の様に割り当てられています．

8インチ両面………サーフェス0，トラック35，セクタ23

5インチ両面………サーフェス1，トラック18，セクタ13

5インチ片面………トラック18，セクタ13

ＩＤセクタには１セクタが割り当てられているので，２５６バイトの領域がある訳ですが，その各々のバイトは，次の様な意味を持っています．

設定ファイル数
ドライブ属性
実行テキスト
(Max. ２５４バイト)

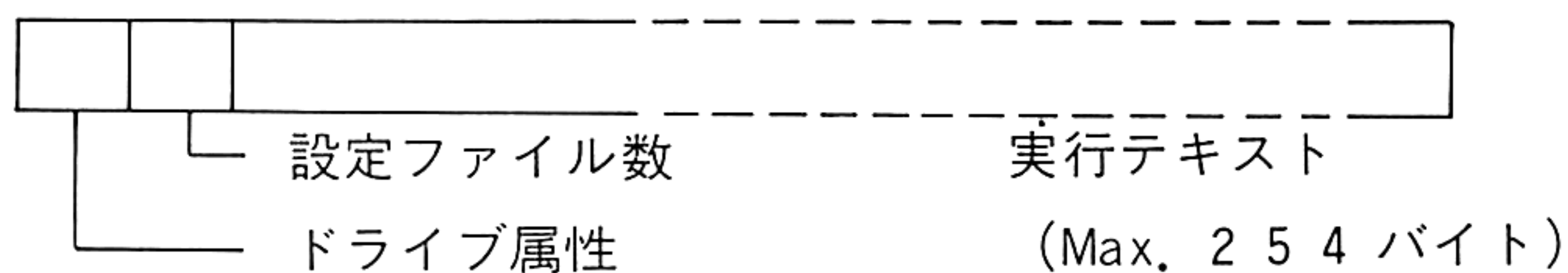



Fig. 1－6　IDセクタ

１バイト目は，そのディスクの持つ属性を示します．以後このディスクに作成されるファイルは，全てこの属性を持つことになります．各ビットの持つ意味はディレクトリのものと同一ですのでそちらを御参照下さい．

２バイト目は，N_{88}－DISK BASICスタート時に設定されるファイルの数を示します．フォーマットしたばかりのシステムディスクではこのバイトにＦＦＨが書き込まれており，ファイルの数をその都度指定することになっています．しかし，このバイトに０～１５の値が書き込まれていた場合には，スタート時に，

How many files（０－１５）?

のプロムプトが表示されず自動的にファイル数の設定が行われます．ただし，この値はシステムディスクでないと意味を持ちません．次のテキストについても同様です．

３バイト目以降はファイル数の設定後に実行されるテキストの内容を示します．そのテキストは２５４文字以内に収まる長さにする必要があります．また，実行させるテキストのない時は，これらのバイトを全て２０Ｈ，もしくは００Ｈで埋め尽くす必要があります．

N_{88}－ＤＩＳＫ ＢＡＳＩＣでは，Ｎ－ＤＩＳＫ ＢＡＳＩＣと異なりファイル数の自動設定を行わなかった場合でも，テキストを自動実行させることができます．では，実際にＩＤセクタをどの様に使用しているかを見てみることにします．

使用したディスクは先のディレクトリ，ＦＡＴの項で使用したものと同一の５インチ両面のものです．

```
Track 12,Surface 01,Sector 0D
0000 40 04 72 75 6E 20 22 64 73 6B 65 64 74 2E 62 61        @ run "dskedt.ba
0010 73 22 00 00 00 00 00 00 00 00 00 00 00 00 00 00        s"
0020 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0030 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0040 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0050 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0060 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0070 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0080 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0090 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00A0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00B0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00C0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00D0 00 00 00 00 00 60 00 00 00 00 00 00 00 00 00 00
00E0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00F0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
```

１バイト目を見ると，４０Ｈが書き込まれており，ビット６が１であることがわかります．これは，このディスクに対しＲ属性が指定されていることを示します．次に２バイト目を見ると，０４Ｈが書き込まれており，N_{88}－ＤＩＳＫ ＢＡＳＩＣスタート時にファイル数を４に設定し，

run "dskedt. bas"

というテキストを実行することを示しています．

1－7　IPL.

N_{88}－ＢＡＳＩＣはシステムのスタート時において，ディスクドライブの接続を確認するとドライブ０，サーフェス０，トラック０（８インチの場合は１），セクタ１より１セクタを本体Ｃ０００Ｈ番地からの領域にロードし，そしてエラーが３回未満であれば，Ｃ０００Ｈ番地に実行を移します．これがＩＰＬで，ディスクコードをメモリ上にロードするのと，ワークエリアをディスクバージョン用にイニシアライズする役目を持っています．

N_{88}－DISK BASICのシステムディスクですと，ディスクコードのロード後８８００Ｈ番地に実行が移されます（このアドレスにはＡＦ００Ｈにジャンプする命令が書き込まれていますので，実際はＡＦ００Ｈに実行が移ります）．このＡＦ００Ｈ番地からのルーチンが，フックの書き換えなどを行うイニシアライズルーチンで，フックの書き換えの後，ＩＤセクタのロード，ファイル数の設定，テキストのキー入力バッファへのアロケートも行います．イニシアライズ終了後はこのルーチンは不用となります（この領域はバッファなどに使用されるため，結局は破壊されることになります）．

また，ディスク上のドライブ０，サーフェス０，トラック０（１），セクタ１の１セクタが無条件にロードされることを利用すると，容易にマシン語レベルでのオートスタートディスクを作成できます．ＣＰ／ＭなどのＯＳも，全てこの最初の１セクタがＩＰＬとなっています．

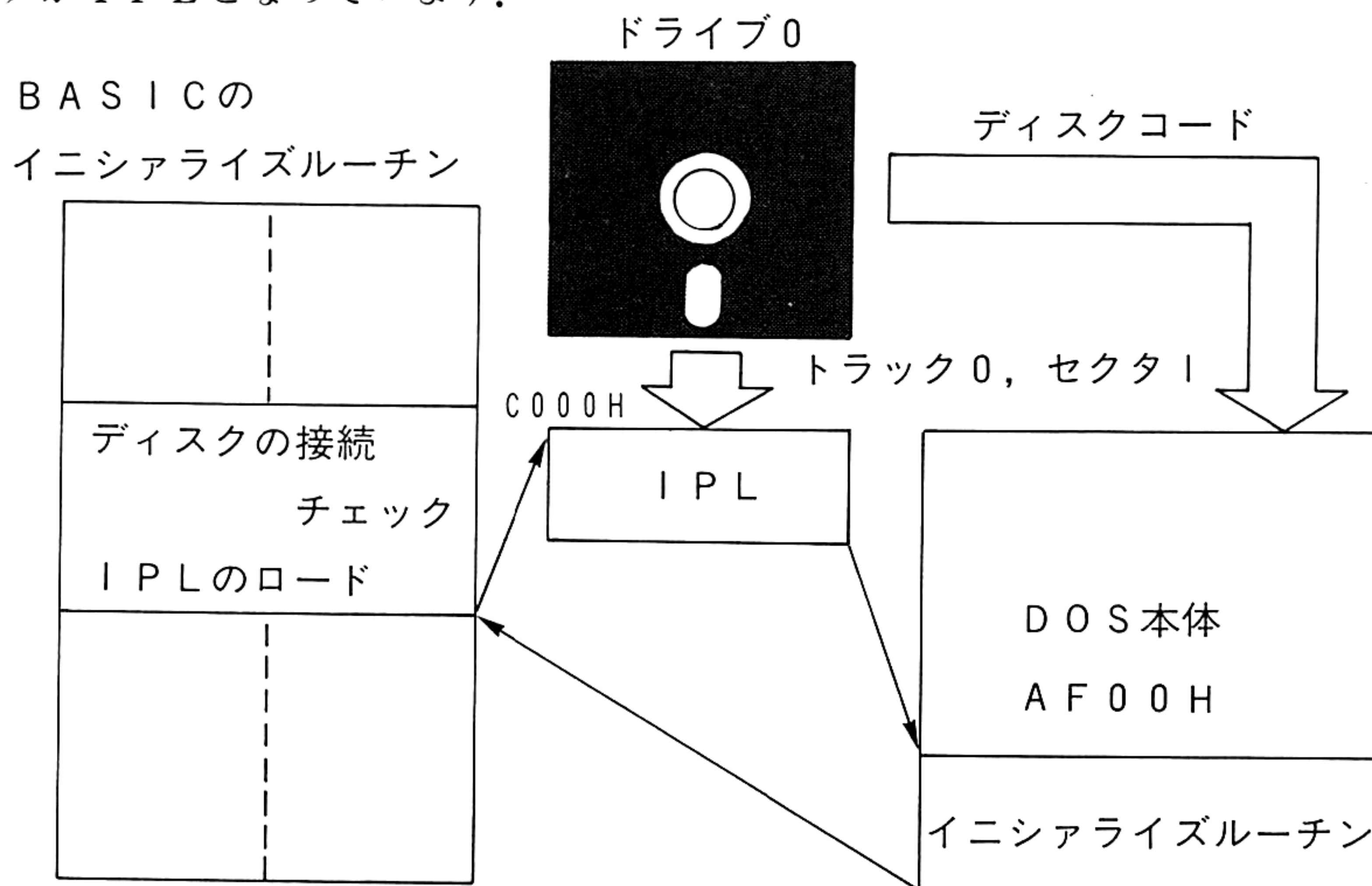



Fig. 1－7　イニシァライズの構造

1－8　ドライブバッファ.

N88－DISK BASICのスタート時に，ドライブバッファが接続されているドライブの数だけメモリ上に確保されます．ドライブバッファというのは，各ドライブのステータス及びFATのコピーを記録してあるもので，そのドライブがマウント状態にあるかなどの情報を与えます．また，ドライブバッファは5バイトのステータスエリアと，160バイトのFATコピーエリアで構成され，全体で1ドライブにつき165バイトの領域を必要とします．

以下に，関連するワークエリア，各ステータス・バイトの設明，ドライブバッファの構造を掲げておきます．

EC81H，82H	バッファのアドレスを並べた表の先頭アドレスを示します．表の長さは（EC7DH）×2バイトです．
EC85H	ドライブ・ナンバの待避用です．ディスクに関するサブルーチンはこの値で示されるドライブに関して動作します．
EC86H，87H	EC85H番地で示されるドライブのバッファ・アドレスが格納されています．ディスクに関するサブルーチンはこの値を常に利用しますので（特にディスクアドレス計算関係），EC85H番地との対応が常になされている必要があります．このアドレスをセットするには，9953H番地からのサブルーチンが最も有効です．
バッファアドレス－2	ディスクのリムーブ時にFATの更新を行なうか否かのフラグです．ファイルへの書き込みが行なわれた場合FFHにセットされます．
バッファアドレス－1	ドライブの属性（IDセクタに書かれているものと同一）を示します．
バッファアドレス＋0	最後にアクセスしたトラック番号を論理トラックで与えます．ディスクが非マウント状態にある場合は，FFHが書き込まれています．またマウント直後はディレクトリトラックの番号が書き込まれています．
バッファアドレス＋1	未使用．
バッファアドレス＋2	ドライブ内のディスクの空きクラスタ数を示します．

バッファアドレス＋３
〜
バッファアドレス＋１６３　そのドライブ内のＦＡＴのコピーです．実際に使用されているのは，８インチ両面で１５４バイト，５インチ両面で１６０バイト，５インチ片面で７０バイトのみです．他の部分は無視されます．

ＥＣ８１Ｈ
ＥＣ８２Ｈ
アドレステーブルの先頭アドレス

ドライブ１
ドライブ２
ドライブｎ
バッファアドレステーブル．ドライブの数だけ確保されます．

－２　ＦＡＴ書き込みフラグ
－１　ドライブの属性
最後にアクセスしたトラック番号．ディスクが非マウント状態のときはＦＦＨ．
＋１　未使用
＋２　空きクラスタ数
＋３　ＦＡＴコピー
〜
＋１６３
ドライブ１

ＥＣ８５Ｈによって示されるドライブのバッファのアドレスを示します．

ＥＣ８６Ｈ
ＥＣ８７Ｈ

ドライブ２
ドライブｎ

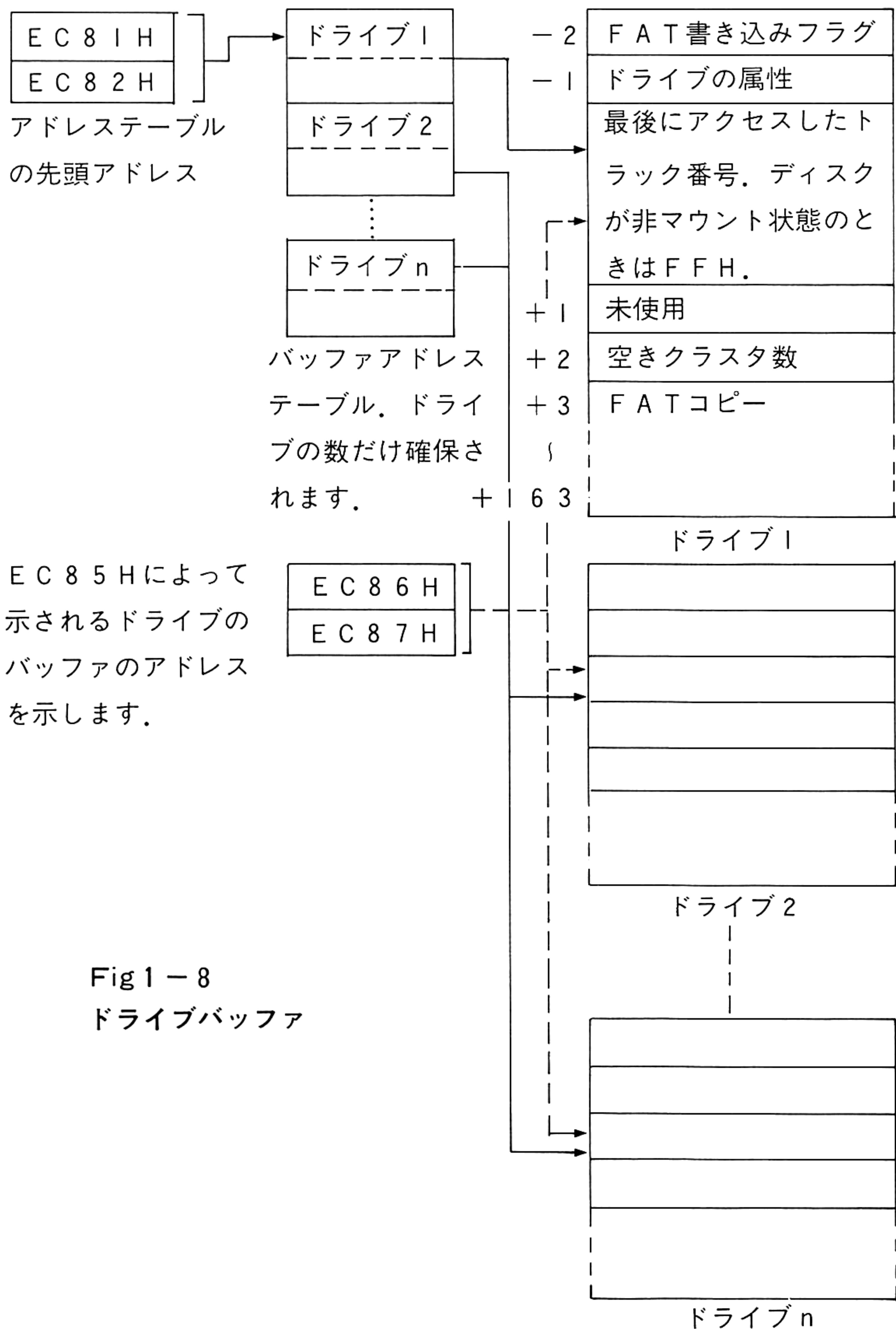



Fig１－８
ドライブバッファ

1－9　ファイルバッファ．

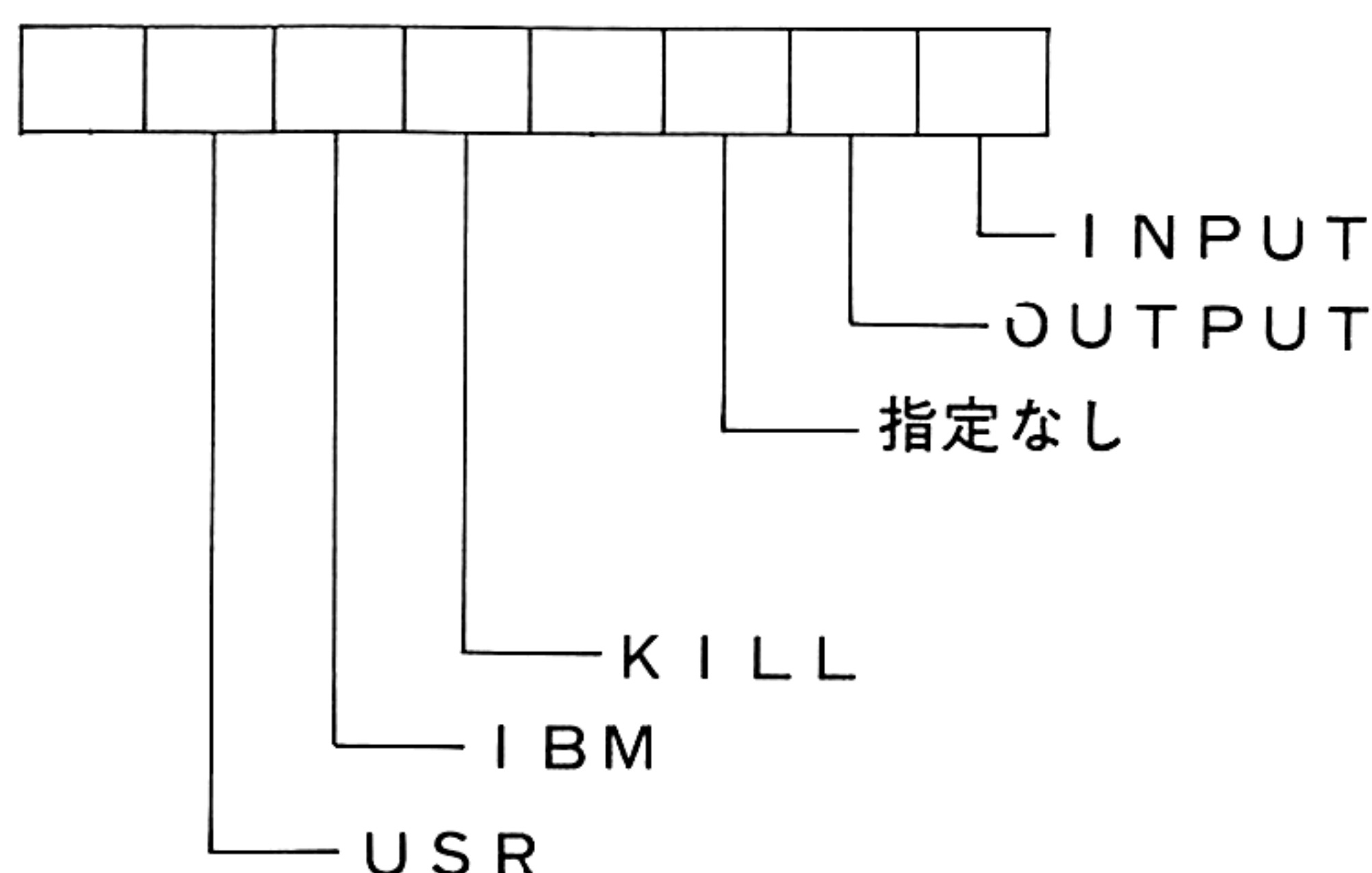
N88－DISK BASICのスタート時に指定した数＋1だけの数のファイルバッファがメモリ上に確保されます．このバッファを通して各周辺デバイスとの入出力が行なわれる訳です（但しRS－232Cとディスクの場合のみで，CMTとキーボードに関しては専用のバッファが用いられます）．

ファイルバッファは9バイトのワークエリア（FCB＝File control block）と256バイトの入出力バッファで構成され，全体で1ファイルにつき265バイトの領域を必要とします．

以下に，関連するワークエリア，FCB各バイトの説明，ファイルバッファの構造を掲げておきます．

EC7FH，80H　バッファのアドレスを並べた表の先頭アドレスを示します．表の長さは（EC7EH）×2バイトです．

EC88H，89H　I／O動作を行うファイルバッファのアドレスを示します．ファイル操作に関するサブルーチンはこのアドレスで示されるファイルに対して行われます．このアドレスのセットには4735H番地からのサブルーチンが最も有効です．

バッファアドレス＋0　ファイルの開かれているモードを示します．各ビットの意味は次の様になります．

このバイトが00Hの時，ファイルはクローズされていることを示します．

バッファアドレス＋1　ファイルの格納されている先頭クラスタを示します．ただし，ディスクファイルでのみ有効です．

バッファアドレス＋２　読み書きを行うクラスタ番号を示します．ただし，ディスクファイルでのみ有効です．

バッファアドレス＋３　読み書きを行なうセクタ番号を示します．ただし，ディスクファイルでのみ有効です．

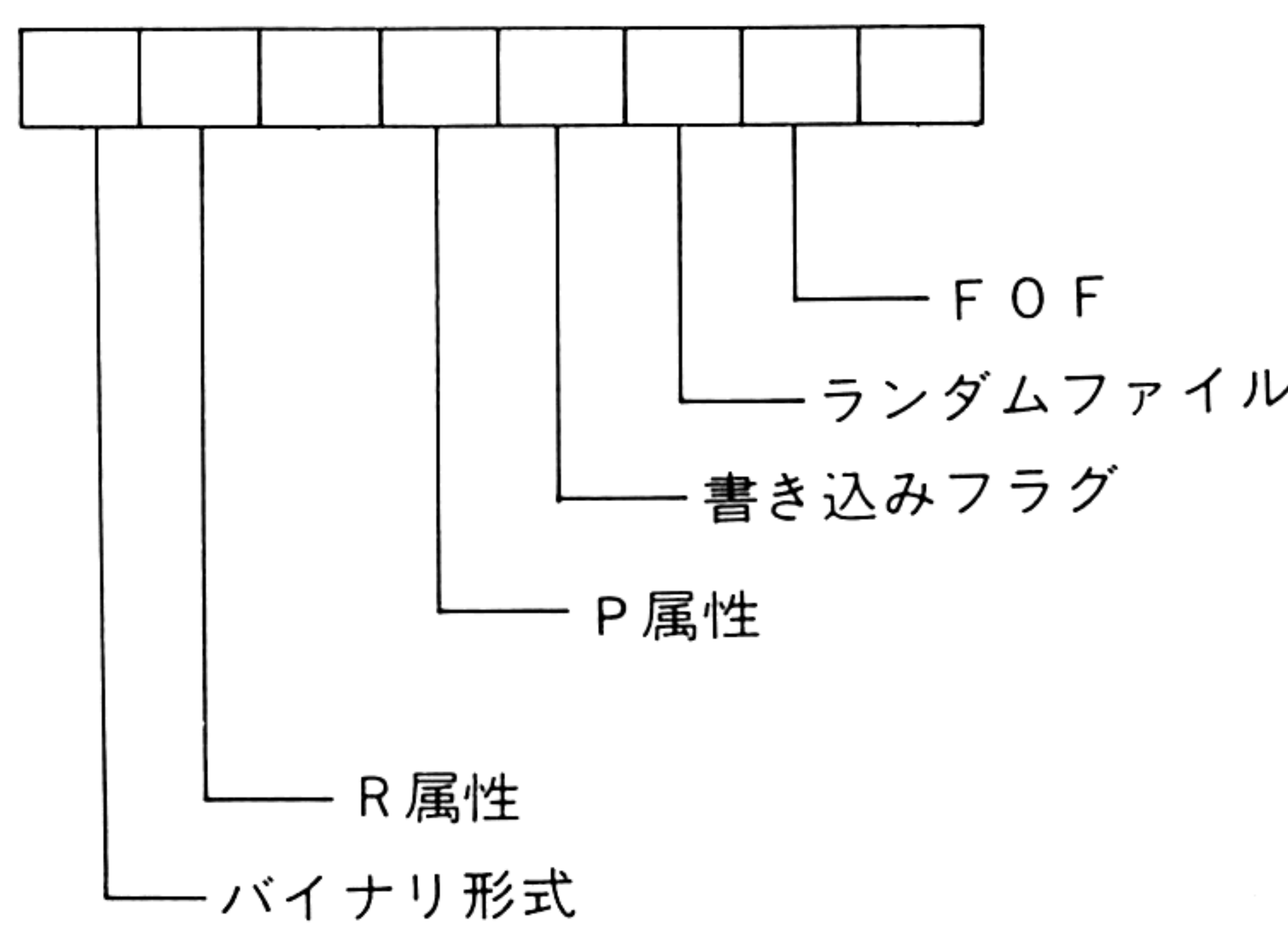
バッファアドレス＋４　ファイルのオープンされているデバイスを示します．デバイス名と番号の関係は次の 様になります．

００Ｈ～０７Ｈ	
	１：～８：
Ｆ８Ｈ	ＳＣＲＮ：
Ｆ９Ｈ	ＬＰＴ１：
ＦＡＨ	ＣＡＳ２：
ＦＢＨ	ＣＡＳ１：
ＦＣＨ	ＣＯＭ３：
ＦＤＨ	ＣＯＭ２：
ＦＥＨ	ＣＯＭ１：
ＦＦＨ	ＫＹＢＤ：

バッファアドレス＋５　バッファの大きさを示します．ＷＩＤＴＨ　＃ｎで指定され，初期値は００Ｈ（２５６）です．

バッファアドレス＋６　レコードのバッファ内における位置をオフセットで示します．レコードとはC_R L_Fで区切られるデータ群のことです．シーケンシャルファイルでのみ有効です．

バッファアドレス＋７　ファイルの属性及びステータスを示します．そのビット構成は次の様になります．

- ＦＯＦ
- ランダムファイル
- 書き込みフラグ
- Ｐ属性
- Ｒ属性
- バイナリ形式

バッファアドレス＋８	１レコード内で次にデータを読み書きする位置を，レコードの先頭からのオフセットで示します．シーケンシャルファイルでのみ有効です．
バッファアドレス＋９	入出力バッファ２５６バイトの先頭です．

EC7FH
EC80H

アドレステーブルの先頭アドレス

ファイル #0
ファイル #1
ファイル #n

バッファアドレステーブル．How many files ？に対して入力した数だけ確保される．

	ファイルモード
+1	先頭クラスタ番号
+2	使用中のセクタ番号
+3	デバイス番号
+4	使用中のクラスタ番号
+5	バッファの大きさ
+6	レコードの先頭
+7	ファイルの属性
+8	データの先頭
+9	ファイルバッファ 256バイト

ファイル#0

I／O動作を行うファイルバッファのアドレス

EC88H
EC89H

ファイル# I

ファイル# n

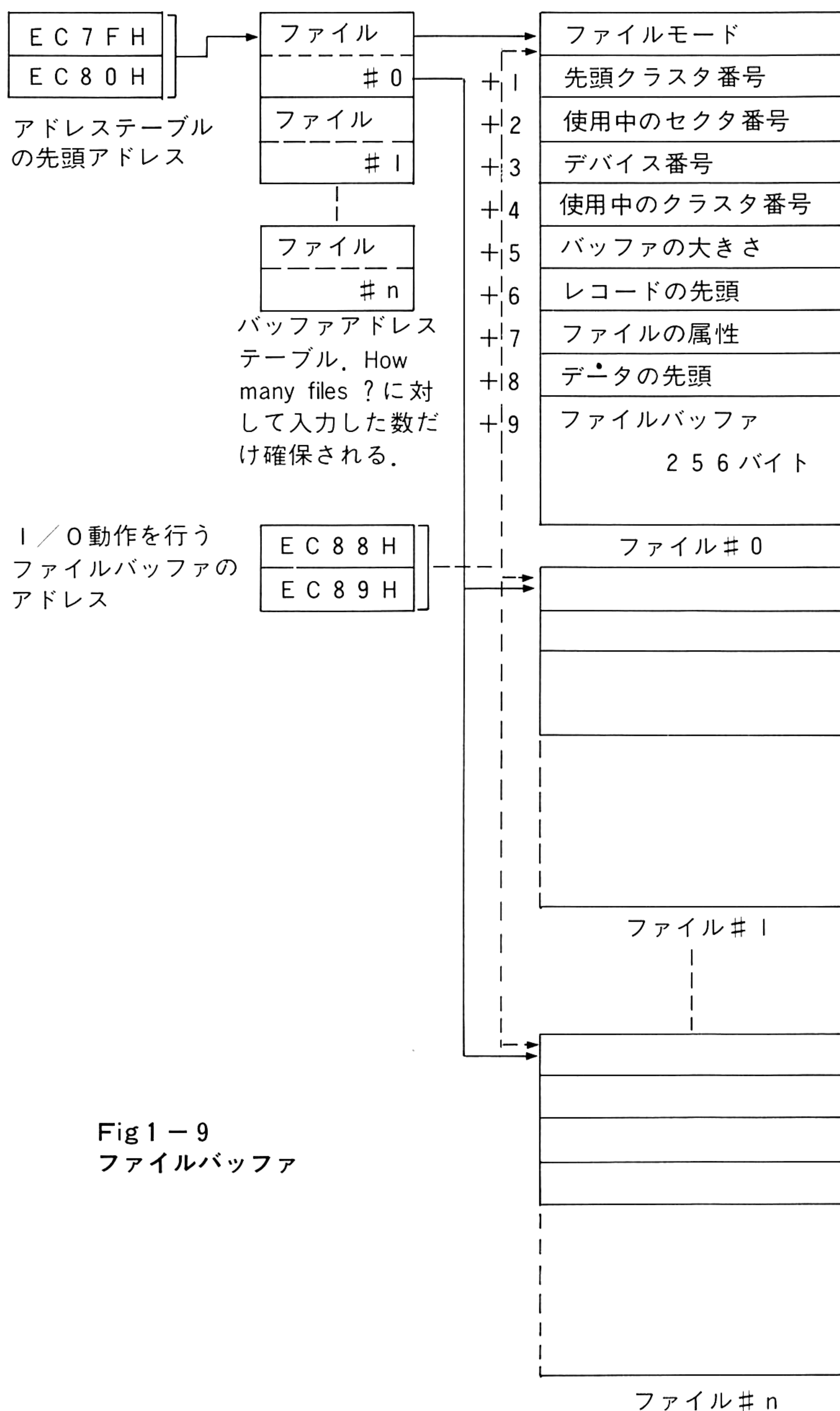



Fig 1－9
ファイルバッファ

1－10　メモリマップ.

N88－DISK BASIC動作時のメモリ・マップを簡単に示します.

領域	アドレス
N88－BASIC インタプリタ	0000H
テキストウィンドウ	8000H
N88－DISK BASIC インタプリタ	8400H
ドライブバッファアドレステーブル	EC81H、82Hで示されるアドレス
ファイルバッファアドレステーブル	EC7FH、80Hで示されるアドレス
ドライブバッファ 1～n	ドライブバッファアドレステーブルの最初の2バイトで示されるアドレス
ファイルバッファ 0～n	ファイルバッファアドレステーブルの最初の2バイトで示されるアドレス
ラベル領域	EB16H、17Hで示されるアドレス
単純変数領域	EB1BH、1CHで示されるアドレス
配列変数領域	EB1DH、1EHで示されるアドレス
フリーエリア	EB1FH、20Hで示されるアドレス
文字列領域	EAF1H、F2Hで示されるアドレス
スタックエリア	EACCH、CDHで示されるアドレス
マシン語領域	E654H、55H（CLEAR文で宣言）で示されるアドレス
ワークエリア	E600H
	FFFFH

Fig. 1－10 メモリアップ

第2章　モニタ(Disk Version)の解析

——Analysis of Monitor(Disk Version)——

2－1　モニタ・ルーチンの利用法．

N88－BASICのマシン語モニタは通常CPUのアドレス空間に現れていません（Fig. 2－1）．そこで，モニタ・ルーチンを利用するには若干の工夫が必要です．

0000H
N－BASIC
インタプリタ
ROM
N88－BASIC
インタプリタ
ROM
6000H
N88－BASIC　モニタ
8000H
8400H
テキスト・ウィンドウ
この他にもN88－BASICの一部が格納されているサブROMが存在しますが，ここでは省きます．
N88－BASIC
メイン　RAM
FFFFH

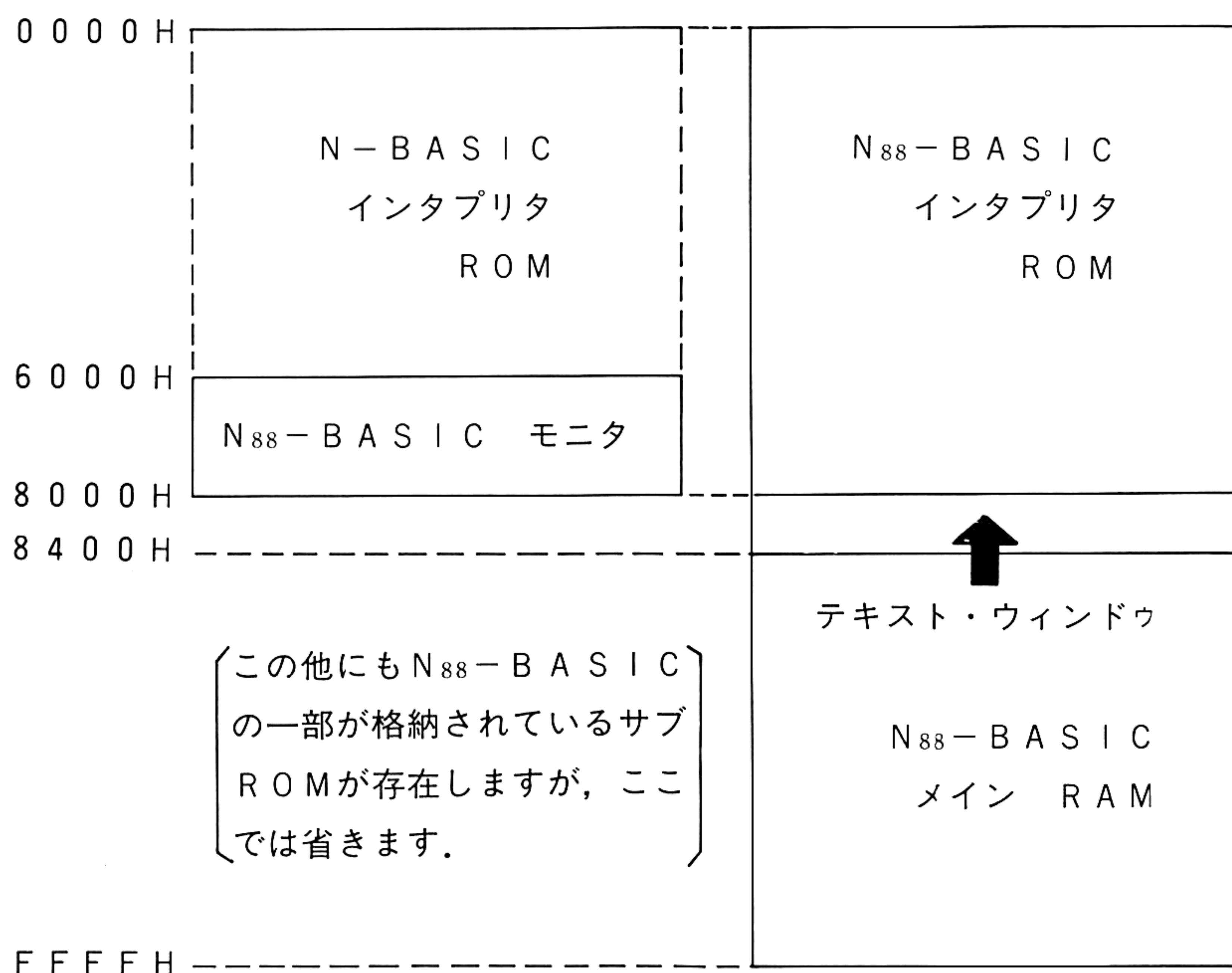



Fig. 2－1　N88－BASIC起動時のメモリ・マップ

Fig. 2－1からわかるように，N88－BASICのモニタはN－BASICインタプリタと同一ROM上にあります．つまりモニタ・ルーチンを呼び出すにはまず，N－BASICモードに切り換えて，それからCALL命令を実行すれば良い訳です．しかしBASICからNEW　ON　1などとするのは不適当です．ソフト的にはポート31Hの第2ビットに1を出力することで可能ですが，他のビットの出力内容を変化させては不都合なので，この点を考慮する必要があります．具体的にはマシン語で，

```
LD      A,(0E6C2H)
OR      04H              ;Aの第2ビットを1にする.
LD      (0E6C2H),A
```

```
        OUT     (31H), A
```
と行えばよい訳です．ここで，E6C2H番地は前にポート31Hに出力した値が格納されているアドレスです．

また，この作業を行なうサブルーチンがRAM上に用意されており，モニタへの切り換えは，

```
        CALL  0F16CH
```
の一命令で可能です．しかし，切り換えたままでは不都合な点が生じるので，元に戻す必要があります．これも，RAM上にサブルーチンが用意されており，

```
        CALL  0F175H
```
の一命令で可能です．

また逆に，モニタモードに切り換えた状態でN_{88}－BASIC－ROMのルーチンをCALLすることも可能です．具体的にはマシン語で，

```
        CALL  0F155H
        DEFW  <アドレス>
```
で実現できます．

なお，ディスク関係のコマンドを処理するルーチンはRAM上にありますが，RAM上のルーチンからROM上のルーチンをCALLしている場合もありますので，RAM上のルーチンを利用する際にもモード切り換えを行なっておくことが必要です．

2－2　モニタ・サブルーチン（ディスク）の解析.

本節では，モニタ・サブルーチンの主にディスクに関するものについて解説していきます．各ルーチンは，エントリ・アドレスの順にまとめることにします．

アドレス　**各ルーチンのエントリ・アドレスです．**

機　　能　**各ルーチンの機能を簡単に示します．**

レジスタ　**各ルーチンの実行後，その内容の変化するレジスタを示します．もし，これらのレジスタの値が変化して不都合の生じる場合などは，その各ルーチンを使用する前にレジスタの退避が必要です．また，N88－DISK　BASIC及びモニタではインデックス・レジスタ（ＩＸ，ＩＹ）及び裏レジスタを使用しておりませんので，それらについては省略します．**

解　　説　各ルーチンの機能を詳しく説明し，入出力パラメータ・注意事項等も示しました．

また，章末にこれらのモニタ・サブルーチンを利用したサンプル・プログラムとして，^D, ^R, ^Wコマンドそれぞれの拡張版及びディスク編集プログラムを掲載しますので，合わせて御参照下さい．

８４００Ｈ　ジャンプテーブルのイニシァライズ.

アドレス　**８４００Ｈ**

機　　能　**モニタのジャンプテーブルをディスク用にイニシァライズする．**

レジスタ　**Ａ，Ｈ，Ｌ**

解　　説　ディスク・バージョンで拡張されたコマンド(^D, ^R, ^W) を使用可能にするために，それぞれのジャンプテーブルを書き換えます．また，ディスク・コマンドに対するＨＥＬＰ (^A) コマンド，ディスク・バージョンで完全となったMコマンド用のジャンプテーブルも書き換えます．

N88－DISK　BASICでは，このルーチンをＩＰＬ以外では使用しておりません．

８４２７Ｈ　モニタ・^Dコマンド処理．

アドレス　８４２７Ｈ

機　能　モニタのＤコマンドの処理を行なう．

レジスタ　Ａ，Ｆ，Ｂ，Ｃ，Ｄ，Ｅ，Ｈ，Ｌ

解　説　モニタの^D（ダンプ・ディスク）コマンドの処理を行ないます．次に，このルーチン内の主な内分けを示します．

８４２７Ｈ－８４６１Ｈ　各パラメータ（ドライブ，サーフェス，トラック，セクタ各ナンバ）の読み込み．

８４６２Ｈ－８４Ｄ８Ｈ　メイン・ルーチン（実際のダンプ）．

８４Ｄ９Ｈ－８４ＥＦＨ　メッセージ・データ．

また，アキュムレータにドライブ・ナンバー１の値をセットし８７９０ＨをＣＡＬＬした後，次のようにパラメータをセットし８４６２Ｈにジャンプすることで，このルーチンを利用できます．

Ｄレジスタ　読み出し開始トラック・ナンバ（論理トラック）．

Ｅレジスタ　読み出し開始セクタ・ナンバ．

Ｈレジスタ　読み出し終了トラック・ナンバ（論理トラック）．

Ｌレジスタ　読み出し終了セクタ・ナンバ．

ここで，トラック・ナンバには論理トラック・ナンバを使用しますので，サーフェスの指定は不必要です．また，ユーザーズ・マニュアルに示される様に，両面にわたるアクセスはできません．

ダンプ終了，もしくはエラーの発生した場合にはモニタに制御が移ります．

８４６２Ｈ番地にジャンプする前にモニタモードへの切り換えが必要です．

８４Ｆ０Ｈ　モニタ・^R，^Wコマンド処理．

アドレス	８４Ｆ０Ｈ
機　能	モニタの^R，^Wコマンドの処理を行う．
レジスタ	A，F，B，C，D，E，H，L

解　説　モニタの^R, ^W（リード・ディスク，ライト・ディスク）コマンドの処理を行ないます．それぞれのエントリ・アドレスは以下の様になります．

８４Ｆ０Ｈ　　^Rコマンド．

８４Ｆ１Ｈ　　^Wコマンド．

次にこのルーチン内の主な内分けを示します．

８４Ｆ２Ｈ－８５３２Ｈ	各パラメータ（ドライブ，サーフェス，トラック，セクタ各ナンバ，開始・終了アドレス）の読み込み
８５３３Ｈ－８５３４Ｈ	^R，^W各処理への分岐．
８５３５Ｈ－８５５ＣＨ	^Rメイン・ルーチン（実際の読み出し）．
８５５ＤＨ－８５８７Ｈ	^Wメイン・ルーチン（実際の書き込み）．

また^Dコマンドと同様に，アキュムレータにドライブ・ナンバー１の値をセットし８７９０ＨをＣＡＬＬした後，次のようにパラメータをセットし８５３３Ｈにジャンプすることでこのルーチンを利用できます．

ＣＹフラグ	^Rの場合０，^Wの場合１．
ＢＣレジスタ・ペア	データ・バイト数．
Ｄレジスタ	アクセスするトラック・ナンバ（論理トラック）．
Ｅレジスタ	アクセスするセクタ・ナンバ．
ＨＬレジスタ・ペア	データ・アドレス．

ここで，トラック・ナンバには論理トラック・ナンバを使用しますので，サーフェスの指定は不必要です．また，ユーザーズ・マニュアルに示される様に，両面にわたるアクセスはできません．

エラーが発生したり，リード・ライトが終了した場合にはモニタに制御が移ります．

８５３３Ｈにジャンプする前にモニタモードへの切り換えが必要です．

８５８８Ｈ　モニタ・各種パラメータ読み込み．

アドレス	８５８８Ｈ
機　　能	**モニタの各種パラメータを読み込む．**
レジスタ	Ａ，Ｆ，Ｈ，Ｌ

解　　説　モニタのディスクコマンドで使用するパラメータ（ドライブ・サーフェス各ナンバ）をキーボードより読み込みます．ただし，ドライブが片面仕様の場合にはドライブ・ナンバのみ読み込みます．

ドライブ・ナンバを読み込み，ドライブの各諸元をセットした後，サーフェスナンバを読み込み値を８６２２Ｈにセットします．片面の場合はＦＦＨがセットされます．

また，エラーが発生するとモニタの６０ＣＥＨ番地に制御が移りますが，ＨＬ＝戻り番地として８５８ＢＨ番地をＣＡＬＬするとその番地に制御が移せます．ただし，スタックが１レベル深くなっているので注意する必要があります．

なお，このサブルーチンからの脱出はコンマ（，）のみです．

このサブルーチンをＣＡＬＬする前にモニタモードへの切り換えが必要です．

８５ＢＡＨ　トラック・ナンバ変換．

アドレス	８５ＢＡＨ
機　　能	**物理トラック・ナンバを論理トラック・ナンバに変換する．**
レジスタ	Ｈ，Ｌ

解　　説　レジスタＬに格納されている物理トラック・ナンバを同じくレジスタＬに論理トラック・ナンバとして返します．

物理・論理トラック・ナンバについては，第１章のトラックの項を御参照下さい．

８５Ｃ９Ｈ　トラック・セクタ正当性チェック．

アドレス	８５Ｃ９Ｈ
機　　能	トラック・セクタ各ナンバの正当性をチェックする．
レジスタ	Ａ，Ｆ，Ｈ，Ｌ

解　　説　トラック・ナンバをレジスタＨに，セクタ・ナンバをレジスタＬにセットしてＣＡＬＬすることにより，そのトラック・セクタ各ナンバが適正な範囲内にあるかどうかチェックします．値が不当な場合はＣＹフラグをセットして戻ります．

ただし，レジスタＨに，与えるトラック・ナンバは，論理トラック・ナンバでなければならず，８６２２Ｈ番地には片面か両面かを与える値（片面の場合ＦＦＨ，両面の場合そのサーフェス・ナンバ）がセットされていなければなりません．また，チェックを行なうドライブの各諸元を，８７９６Ｈ番地からのサブルーチンによってあらかじめセットしておく必要があります．

８５ＤＢＨ　トラック・セクタ・インクリメント．

アドレス	８５Ｃ９Ｈ
機　　能	トラック・セクタ各ナンバを増加させる．
レジスタ	Ａ，Ｆ，Ｈ，Ｌ

解　　説　トラック・ナンバをレジスタＨに，セクタ・ナンバをレジスタＬにセットしてＣＡＬＬすることにより，セクタ・ナンバをプラス１し，その結果が最大セクタ・ナンバを越えた場合にはさらにトラック・ナンバをプラス１し，セクタ・ナンバに１を入れた値を返します．その結果が不当な場合はＣＹフラグをセットして戻ります．

ただし，レジスタＨに与えるトラック・ナンバは論理トラック・ナンバでなければならず，８６２２Ｈ番地には両面か片面かを与える値（片面の場合ＦＦＨ，両面の場合そのサーフェス・ナンバ）がセットされていなければなりません．また，チェックを行なうドライブの各諸元を，８７９６Ｈ番地からのサブルーチンによってあらかじめセットしておく必要があります．

８５ＥＤＨ　モニタ・ディスク入出力（Ｉ）．

アドレス	８５ＥＤＨ
機　能	**モニタモードでディスクと１セクタの入出力を行う．**
レジスタ	Ａ，Ｆ，Ｂ，Ｃ，Ｄ，Ｅ，Ｈ，Ｌ

解　説　モニタモードにおいて，ディスクとの１セクタの入出力行なうときに使用するサブルーチンです．リードのときは８５ＥＤＨ番地，ライトのときは８５ＥＥＨ番地にそれぞれエントリします．そのときに与えるパラメータは，

ＨＬレジスタペア	データ・アドレス．
Ｄレジスタ	トラック・ナンバ．
Ｅレジスタ	セクタ・ナンバ．
ＥＣ８５Ｈ	ドライブ・ナンバー１．
ＥＦ５ＤＨ	ドライブ・タイプ（０～３）．

　となります．ここで，トラック・ナンバには論理トラック・ナンバを使用しますので，サーフェスの指定は不必要です．エラー発生の場合，モニタのエラー処理ルーチン（６０ＣＥＨ番地）に制御が移ります．このサブルーチンをＣＡＬＬする前にモニタモードの切り換えが必要です．

８５ＦＣＨ　ディスク入出力（II）.

アドレス	８５ＦＣＨ
機　　能	ディスクと１セクタの入出力を行う.
レジスタ	Ｆ

解　　説　ディスクと１セクタの入出力を行ないます．モニタでは８５ＥＤＨ番地からのサブルーチンで使用されているのみです．与えるパラメータは，

ＣＹフラグ	ライト……１，リード……０．
Ｚフラグ	ベリファイ……１，リード……０．
Ｄレジスタ	トラック・ナンバ．
Ｅレジスタ	セクタ・ナンバ．
ＨＬレジスタペア	データ・アドレス．
ＥＣ８５Ｈ	ドライブ・ナンバー１．
ＥＦ５ＤＨ	ドライブ・タイプ（０～３）．

となります．ここで，トラック・ナンバには論理トラック・ナンバを使用しますので，サーフェスの指定は不必要です．また，トラック・ナンバとセクタ・ナンバの指定方法が３６９ＡＨ番地からのサブルーチンと異なるので御注意下さい．エラーが発生した場合には，ＣＹフラグをセットして戻ります．

8790H　モニタ・ドライブIDセット.

アドレス	8790H
機　能	モニタモードにおいてドライブの各諸元をセットする.
レジスタ	全て退避

解　説　モニタより，指定ドライブの各諸元をセットするときに使用します．パラメータとしてはアキュムレータにドライブ・ナンバー1の値をセットしますが，得られるデータは4742H番地からのサブルーチンに準じるのでそちらを御参照下さい.

具体的には，F155Hからのサブルーチンを利用して4742H番地を間接的にCALLしています.

8796H　モニタ・ドライブID，アドレスセット.

アドレス	8796H
機　能	モニタモードにおいて，ドライブアドレスをセットする.
レジスタ	A，F，D，E，H，L

解　説　モニタより，指定ドライブの各諸元と，ドライブアドレスをセットするときに使用します．与えるパラメータとしてはアキュムレータにドライブ・ナンバの値をセットしますが，得られるデータは9953H番地からのサブルーチンに準じるので，そちらを御参照下さい.

具体的には，F155Hからのサブルーチンを利用して9953H番地を間接的にCALLしています.

ただし，アキュムレータに与えた値が0以下もしくは接続されているドライブ数を越えた場合にはエラーとなります.

2－3　サンプル・プログラム.

本節では，モニタ・サブルーチンを利用したプログラムを二種紹介します.

第1のプログラムは基本的には^D，^R，^W各コマンドと同じですが，モニタが入力すべき内容を指示しますので，入力が確実に行えます．また，ディスクアドレスの他にクラスタ番号による指定もできますので，ファイルの内容を調べることも容易です．しかし，モニタ・ルーチンをそのまま利用しているため，ユーザーズ・マニュアルにもあるようにディスクの両面にまたがってアクセスすることはできません.

第2のプログラムはディスクのエディタ・プログラムです．ルーチンのほとんどはモニタのＥコマンドのサブルーチンを利用しています．起動はＣＴＲＬ＋Ｅとタイプし，パラメータの指定方法は^Ｄコマンドと同様です.

プログラムは,全てマシン語で記述されており，ＰＣ－８８０１の内蔵モニタより入力可能ですが，アセンブラがザイログ仕様のため内蔵アセンブラ（インテル仕様）では直接入力できません．そこで，入力の際にはＡＰＰＥＮＤＩＸのμＣＯＭ－８２・インテル・ニモニック対応表を利用するか，モニタのＥコマンドあるいはＳコマンドでオブジェクト・コードを直接入力してください.

また，プログラムをモニタより呼び出す前に,ポインタの書き換えが必要となるので，以下の様に変更して下さい.

Ｅ６７４Ｈ，Ｅ６７７Ｈ，Ｅ６７ＡＨ→Ｅ４Ｈ
Ｅ６７３Ｈ→６ＢＨ
Ｅ６７６Ｈ→４１Ｈ
Ｅ６７９Ｈ→４０Ｈ

Ｅ６７０Ｈ→Ｂ０Ｈ
Ｅ６７１Ｈ→Ｅ４Ｈ

注） ２つのプログラムを同時にメモリ上に置くことはできませんので御了承下さい.

サンプル

```
                 ;
                 ; monitor extended disk command (^r,^w,^d)
                 ;
                 ; set E674H,E677H and E67AH to E4H
                 ;     E673H to 6BH,E676H to 41H,E679H to 40H
                 ;
                         ORG   0E440H
                 ;
                 ; read/write command entry
                 ;
E440 F637        MCOMRW:OR    37H
E442 F5                 PUSH  AF               ; save CY flag
E443 CD8FE4             CALL  INPDRV           ; input drive number
                 ;
E446 CDACE4             CALL  CHOIC1           ; cluster or track,sector
E449 D4D9E4             CALL  NC,INPCLS
E44C DCFFE4             CALL  C,INPTRK
                 ;
E44F D5                 PUSH  DE
E450 21DCE5      MCMRW1:LD    HL,TOPADR        ; input top address
E453 CD40E5             CALL  INPADR
E456 EB                 EX    DE,HL
E457 21E9E5             LD    HL,ENDADR
E45A CD40E5             CALL  INPADR
E45D AF                 XOR   A                ; clear CY flag
E45E ED52               SBC   HL,DE
E460 38EE               JR    C,MCMRW1
E462 44                 LD    B,H              ; BCreg=length of data
E463 4D                 LD    C,L
E464 03                 INC   BC
E465 EB                 EX    DE,HL            ; HLreg=top of data
E466 D1                 POP   DE
                 ;
E467 F1                 POP   AF
E468 C33385             JP    8533H            ; entry to read/write main
                 ;
                 ; dump command entry
                 ;
E46B AF          MCOMDP:XOR   A
E46C 32DFF1             LD    (0F1DFH),A       ; acceptable control key (^s...)
E46F CD8FE4             CALL  INPDRV
                 ;
E472 CDACE4             CALL  CHOIC1
E475 D4D9E4             CALL  NC,INPCLS
E478 DCFFE4             CALL  C,INPTRK
                 ;
E47B CDC2E4             CALL  CHOIC2
E47E 62                 LD    H,D
E47F 6B                 LD    L,E
E480 300A               JR    NC,MCMDP1        ; only one cluster skip
                 ;
E482 E5                 PUSH  HL
E483 79                 LD    A,C
E484 A7                 AND   A
E485 C4D9E4             CALL  NZ,INPCLS
E488 CC06E5             CALL  Z,INPTR1
E48B E1                 POP   HL
                 ;
E48C C36184      MCMDP1:JP    8461H            ; entry to dump main
                 ;
                 ; subroutine input drive number
                 ;
E48F 2170E5      INPDRV:LD    HL,DRVNUM        ; input drive number
E492 CD2E70             CALL  702EH            ; output strings until (HL)=0
E495 CD56E5             CALL  INPPRM
E498 DA8860             JP    C,6088H          ; monitor main loop
                 ;
E49B 47                 LD    B,A
E49C 3A7DEC             LD    A,(0EC7DH)       ; maximum drive number
E49F 05                 DEC   B
E4A0 FA8FE4             JP    M,INPDRV
```

```
E4A3 3D               DEC  A
E4A4 B8               CP   B
E4A5 38E8             JR   C,INPDRV
E4A7 78               LD   A,B
E4A8 CD9087           CALL 8790H                ; set disk information
E4AB C9               RET
               ;
               ; subroutine choose way of input parameter
               ;
E4AC 217BE5    CHOIC1:LD   HL,CHCMS1
E4AF CD2E70           CALL 702EH
E4B2 CD55F1           CALL 0F155H
E4B5 C85F             DEFW 5FC8H
E4B7 23               INC  HL
E4B8 7E               LD   A,(HL)
E4B9 FE63             CP   'c'                  ; cluster ?
E4BB C8               RET  Z                    ; carry flag=0
E4BC FE74             CP   't'                  ; track,sector
E4BE 3F               CCF
E4BF C8               RET  Z                    ; carry flag=1
E4C0 18EA             JR   CHOIC1
               ;
               ; subroutine choose number of parameter
               ;
E4C2 D5        CHOIC2:PUSH DE
E4C3 C5               PUSH BC
E4C4 219CE5           LD   HL,CHCMS2
E4C7 CD2E70           CALL 702EH
E4CA CD55F1           CALL 0F155H
E4CD C85F             DEFW 5FC8H
E4CF 23               INC  HL
E4D0 7E               LD   A,(HL)
E4D1 FE31             CP   '1'
E4D3 2801             JR   Z,CHOICJ             ; carry flag=0
E4D5 37               SCF                       ; else carry flag=1
E4D6 C1        CHOICJ:POP  BC
E4D7 D1               POP  DE
E4D8 C9               RET
               ;
               ; subroutine input cluster number
               ;
E4D9 F5        INPCLS:PUSH AF
E4DA 21C1E5    INPCL1:LD   HL,CLSNUM            ; input cluster number
E4DD CD2E70           CALL 702EH
E4E0 CD56E5           CALL INPPRM
E4E3 47               LD   B,A
               ;
E4E4 CD21E5           CALL INPSEC               ; input sector number
E4E7 4F               LD   C,A
E4E8 CD55F1           CALL 0F155H
E4EB FFA4             DEFW 0A4FFH               ; convert cluster to track
E4ED 78               LD   A,B
E4EE E601             AND  1                    ; surface number
E4F0 322286           LD   (8622H),A
E4F3 69               LD   L,C
E4F4 60               LD   H,B
E4F5 CDC985           CALL 85C9H                ; check track and sector
E4F8 38E0             JR   C,INPCL1
E4FA EB               EX   DE,HL
E4FB F1               POP  AF
E4FC 0E01             LD   C,1                  ; cluster input flag
E4FE C9               RET
               ;
               ; subroutine input track and sector number
               ;
E4FF 3AA7EC    INPTRK:LD   A,(0ECA7H)           ; surface flag
E502 A7               AND  A
E503 C42FE5           CALL NZ,INPSUR
               ;
E506 21CAE5    INPTR1:LD   HL,TRKNUM            ; input track number
E509 CD2E70           CALL 702EH
E50C CD56E5           CALL INPPRM
E50F 6F               LD   L,A
E510 CDBA85           CALL 85BAH                ; convert physical to logical
```

```
                    ;
E513 CD21E5                CALL  INPSEC              ; input sector number
E516 65                    LD    H,L
E517 6F                    LD    L,A
E518 CDC985                CALL  85C9H               ; check track and sector
E51B 38E9                  JR    C,INPTR1
E51D EB                    EX    DE,HL
E51E 0E00                  LD    C,0                 ; disk address flag
E520 C9                    RET
                    ;
                    ; subroutine input sector number
                    ;
E521 C5             INPSEC:PUSH BC
E522 E5                    PUSH  HL
E523 21D3E5                LD    HL,SECNUM
E526 CD2E70                CALL  702EH
E529 CD56E5                CALL  INPPRM
E52C E1                    POP   HL
E52D C1                    POP   BC
E52E C9                    RET
                    ;
                    ; subroutine input surface number
                    ;
E52F 21B8E5         INPSUR:LD    HL,SURNUM
E532 CD2E70                CALL  702EH
E535 CD56E5                CALL  INPPRM
E538 FE02                  CP    2
E53A 30F3                  JR    NC,INPSUR           ; over 2
E53C 322286                LD    (8622H),A
E53F C9                    RET
                    ;
                    ; subroutine input address
                    ;
E540 D5             INPADR:PUSH DE
E541 CD2E70                CALL  702EH
E544 CD75F1                CALL  0F175H              ; select main ROM routine
E547 CDC85F                CALL  5FC8H               ; line input
E54A 23                    INC   HL
E54B CDBC26                CALL  26BCH
E54E CDA021                CALL  21A0H
E551 CD6CF1                CALL  0F16CH              ; select alternative ROM routine
E554 D1                    POP   DE
E555 C9                    RET
                    ;
                    ; subroutine input parameter
                    ;
E556 D5             INPPRM:PUSH DE
E557 CD75F1         INPPR1:CALL 0F175H
E55A CDC85F                CALL  5FC8H
E55D 380C                  JR    C,INPPR2            ; stop key return with carry
E55F 23                    INC   HL
E560 CDBC26                CALL  26BCH               ; convert to binary
E563 CDA021                CALL  21A0H               ; convert to integer
E566 7C                    LD    A,H
E567 B7                    OR    A
E568 20ED                  JR    NZ,INPPR1           ; over 255
E56A 7D                    LD    A,L
E56B CD6CF1         INPPR2:CALL 0F16CH
E56E D1                    POP   DE
E56F C9                    RET
                    ;
                    ; message area
                    ;
E570 0D0A4472       DRVNUM:DEFM 0DH,0AH,'Drive  :',0
E574 69766520
E578 203A00
E57B 436C7573       CHCMS1:DEFM 'Cluster(c) or '
E57F 74657228
E583 6329206F
E587 7220
E589 54726163              DEFM  'Track,Sector(t) ? ',0
E58D 6B2C5365
E591 63746F72
E595 28742920
```

```
E599 3F2000
E59C 4F6E6C79 CHCMS2:DEFM 'Only one sector (1/else) ? ',0
E5A0 206F6E65
E5A4 20736563
E5A8 746F7220
E5AC 28312F65
E5B0 6C736529
E5B4 203F2000
E5B8 53757266 SURNUM:DEFM 'Surface:',0
E5BC 6163653A
E5C0 00
E5C1 436C7573 CLSNUM:DEFM 'Cluster:',0
E5C5 7465723A
E5C9 00
E5CA 54726163 TRKNUM:DEFM 'Track  :',0
E5CE 6B20203A
E5D2 00
E5D3 53656374 SECNUM:DEFM 'Sector :',0
E5D7 6F72203A
E5DB 00
E5DC 546F7020 TOPADR:DEFM 'Top address:',0
E5E0 61646472
E5E4 6573733A
E5E8 00
E5E9 456E6420 ENDADR:DEFM 'End address:',0
E5ED 61646472
E5F1 6573733A
E5F5 00
                 ;
E5F6                    END
```

```
clear ,&he43f
Ok
bload "mexcom",&he440
Ok
poke &he674,&he4:poke &he677,&he4:poke &he67a,&he4
Ok
poke &he673,&h6b:poke &he676,&h41:poke &he679,&h40
Ok
mon

hJ^d

Drive  :1
Cluster(c) or Track,Sector(t) ? c
Cluster:0
Sector :1
Only one sector (1/else) ? 1

Track 00,Surface 00,Sector 01
0000 21 00 84 AF 32 B4 EC 01 02 00 11 11 2F 3A 5D EF     !  ■ッ2エ●       /:コ\
0010 B7 20 04 06 02 1E 1B AF 3C E5 D5 C5 CD 9A 36 C1     キ        ッく◣ユナヘ└6チ
0020 D1 E1 30 08 3A B4 EC FE 03 20 EC C9 AF 32 B4 EC     ム ト0  :エ●    ●ノッ2エ●
0030 15 28 0B 0C 24 7B B9 20 DE 04 0E 01 18 D9 CD 00      (   ${ケ "       ルヘ
0040 84 3E 30 D3 F3 21 FF 7F 2B 7C B5 20 FB 3E 00 D3     ■>0モ月!   +¦オ   > モ
0050 F3 C3 00 88 00 00 00 C3 23 71 23 77 EB 10 EE C9     月テ |     テ#q#w♣ //
0060 DC EC AC D9 C4 EC A2 D0 C1 EC DE D9 D0 EC 14 D4     ワ●ャルト●「ミチ●"ルミ● ヤ
0070 AE ED 8E D9 E8 EC A4 D4 B1 ED 4A CE EE EC EB D4     ョ○■ル♣●、ヤアOJホ/●♣ャ
0080 F1 EC 6F D9 F4 EC 82 D9 6F ED FE CF AB ED E1 CE     円●oル日●_ルoO マォO卜ホ
0090 FF ED 76 CE BB EC 4B C3 D3 EC D8 ED D6 EC D8 ED      Ovホサ●Kテモ●リOヨ●リO
00A0 9F ED D8 ED A2 ED D8 ED A2 ED A5 ED A8 ED D8 ED     ノOリO「OリO「O・OィOリO
00B0 74 EE D8 ED 77 EE D8 ED 7A EE E0 DB 7D EE 22 C2     t/リOw/リOz/=□}/"ツ
00C0 80 EE CC DA 83 EE F6 DA 86 EE A5 DE 89 EE 6D DB     _/フレ■/分レ■/・"| /m□
00D0 8C EE 56 D2 8F EE 96 CE 92 EE 8A D0 95 EE 08 D5     |/Vメ+/|ホ┤/|ミー/ ユ
00E0 98 EE 36 D5 9B EE 49 D3 9E EE 44 D3 BC EE 00 C1     r/6ユ┘/Iモ\/□モシ/ チ
00F0 BF EE 23 DA C2 EE E7 D9 C5 EE A2 DE C8 EE EC C2     ソ/#レツ/◤ルナ/「"ネ/●ツ
hJ^b
Ok
```

サンプル

```
                  ;
                  ; monitor disk edit program
                  ;
                  ;   to execute,key in 'CTRL+E'
                  ;
                  ;   set E670H to B0H,E671H to E4H
                  ;
                         ORG   0E4B0H
                  ;
E4B0 FE4D         DSKED: CP    'M'              ; M command
E4B2 CA6D87              JP    Z,876DH
                  ;
E4B5 FE05                CP    'E'-40H          ; non 'CTRL+E' return
E4B7 C0                  RET   NZ
                  ;
E4B8 F1                  POP   AF               ; return cancel
                  ;
E4B9 3AB0E6              LD    A,(0E6B0H)       ; upper range for scroll
E4BC 3D                  DEC   A
E4BD C0                  RET   NZ
                  ;
E4BE 3AB1E6              LD    A,(0E6B1H)       ; lower range for scroll
E4C1 FE15                CP    15H
E4C3 D8                  RET   C
                  ;
E4C4 3A89EF              LD    A,(0EF89H)
E4C7 FE29                CP    40+1
E4C9 D8                  RET   C                ; 40 mode then error
                  ;
E4CA 3EFF                LD    A,0FFH
E4CC 32DFF1              LD    (0F1DFH),A       ; clear printer flag
                  ;
E4CF CD8885              CALL  8588H            ; input drive and surface number
                  ;
E4D2 CDEE6E       DSKED1:CALL  6EEEH            ; input track number
E4D5 CDBA85              CALL  85BAH            ; convert to logical track
E4D8 4D                  LD    C,L
E4D9 D8                  RET   C
E4DA FE2C                CP    ','
E4DC C0                  RET   NZ
                  ;
E4DD CDEE6E              CALL  6EEEH            ; input sector number
E4E0 61                  LD    H,C
E4E1 D8                  RET   C
E4E2 FE0D                CP    0DH
E4E4 C0                  RET   NZ
                  ;
E4E5 54                  LD    D,H
E4E6 5D                  LD    E,L
E4E7 CDC985              CALL  85C9H            ; check track and sector
E4EA D8                  RET   C
                  ;
E4EB D5                  PUSH  DE
E4EC 3A85EC              LD    A,(0EC85H)
E4EF 3C                  INC   A
E4F0 CD9687              CALL  8796H            ; set disk identification
E4F3 D1                  POP   DE
E4F4 2ADDE5              LD    HL,(BUFFER)      ; top of edit buffer
                  ;
E4F7 CDED85              CALL  85EDH            ; read disk
                  ;
E4FA 3E0C         DSKED2:LD    A,12             ; clear CRT
E4FC CD2971              CALL  7129H
                  ;
E4FF D5                  PUSH  DE               ; track and sector save
E500 E5                  PUSH  HL               ; address save
E501 1100FF              LD    DE,0FF00H
                  ;
E504 CDD76F       DSKED3:CALL  6FD7H            ; compare DE and HL
E507 3801                JR    C,DSKED4
E509 EB                  EX    DE,HL
                  ;
```

```
E50A E5        DSKED4:PUSH HL
E50B CD3D67           CALL 673DH            ; dump
E50E D1               POP  DE
E50F E1               POP  HL
E510 CDF666           CALL 66F6H
              ;
E513 D1               POP  DE
E514 CD87E5           CALL IDDSP            ; display each number
              ;
E517 CD7B71    DSKED5:CALL 717BH            ; cursor on
E51A E5               PUSH HL
E51B 2117E5           LD   HL,DSKED5
E51E E3               EX   (SP),HL
E51F CD5971           CALL 7159H            ; key input
              ;
E522 E5        DSKED6:PUSH HL
E523 C5               PUSH BC
E524 213BE5           LD   HL,COMTBL        ; command table
E527 010800           LD   BC,08H
E52A EDB1             CPIR
E52C C26265           JP   NZ,6562H
E52F 2143E5           LD   HL,ADRTBL        ; address table
E532 09               ADD  HL,BC
E533 09               ADD  HL,BC
E534 C1               POP  BC
E535 7E               LD   A,(HL)
E536 23               INC  HL
E537 66               LD   H,(HL)
E538 6F               LD   L,A
E539 E3               EX   (SP),HL
E53A C9               RET                   ; execusion of command
              ;
E53B 1E1F1D1C COMTBL:DEFB 1EH,1FH,1DH,1CH   ; cursor move
E53F 031B0602         DEFB 03H,1BH,06H,02H  ; exit and cursor skip
              ;
E543 C465      ADRTBL:DEFW 65C4H            ; CTRL+F
E545 BD65             DEFW 65BDH            ; CTRL+B
E547 53E5             DEFW EXIT             ; ESC
E549 53E5             DEFW EXIT             ; STOP,CTRL+C
E54B 1B66             DEFW 661BH            ; cursor right
E54D 2F66             DEFW 662FH            ; cursor left
E54F EC65             DEFW 65ECH            ; cursor down
E551 CB65             DEFW 65CBH            ; cursor up
              ;
E553 211401    EXIT:  LD   HL,0114H
E556 2286EF           LD   (0EF86H),HL
E559 21E5E5           LD   HL,QUEST         ; exit command
E55C CD4C70           CALL 704CH
E55F CD5971           CALL 7159H            ; input 1 charactor
E562 CD2971           CALL 7129H
E565 CD3770           CALL 7037H            ; line feed
E568 CD2570           CALL 7025H            ; convert to large
              ;
E56B ED5B1EF2         LD   DE,(0F21EH)      ; load track and sector
E56F 2ADDE5           LD   HL,(BUFFER)
              ;
E572 FE43             CP   'C'
E574 CA8860           JP   Z,6088H
E577 FE4E             CP   'N'
E579 C1               POP  BC
E57A CAFAE4           JP   Z,DSKED2
E57D FE59             CP   'Y'
E57F 20D2             JR   NZ,EXIT
              ;
E581 CDEE85           CALL 85EEH            ; write disk
              ;
E584 C38860           JP   6088H
              ;
E587 E5        IDDSP: PUSH HL
E588 E5               PUSH HL
E589 C5               PUSH BC
E58A ED531EF2         LD   (0F21EH),DE
E58E 211201           LD   HL,0112H         ; cursor set
E591 2286EF           LD   (0EF86H),HL
```

```
E594 21DFE5            LD   HL,DRIVE
E597 CD4C70            CALL 704CH
E59A 3A85EC            LD   A,(0EC85H)
E59D 3C                INC  A
E59E CD806F            CALL 6F80H            ; 1 byte number output
                ;
E5A1 3E2C              LD   A,','
E5A3 CD2971            CALL 7129H
E5A6 21D984            LD   HL,84D9H         ; top of message ',Track'
E5A9 CD4C70            CALL 704CH
E5AC 3A2286            LD   A,(8622H)
E5AF 87                ADD  A,A
E5B0 7A                LD   A,D
E5B1 3801              JR   C,IDDSP1
E5B3 1F                RRA
E5B4 CD806F     IDDSP1:CALL 6F80H
E5B7 3A2286            LD   A,(8622H)
E5BA 87                ADD  A,A
E5BB 380C              JR   C,IDDSP2
                ;
E5BD 21DF84            LD   HL,84DFH         ; top of message ',Surface'
E5C0 CD4C70            CALL 704CH
E5C3 7A                LD   A,D
E5C4 E601              AND  01H
E5C6 CD806F            CALL 6F80H
                ;
E5C9 21E884     IDDSP2:LD   HL,84E8H         ; top of message ',Sector'
E5CC CD4C70            CALL 704CH
E5CF 7B                LD   A,E
E5D0 CD806F            CALL 6F80H
                ;
E5D3 210106            LD   HL,0601H         ; cursor home
E5D6 2286EF            LD   (0EF86H),HL
E5D9 C1                POP  BC
E5DA E1                POP  HL
E5DB D1                POP  DE
E5DC C9                RET
                ;
E5DD B9E9       BUFFER:DEFW 0E9B9H
                ;
E5DF 44726976 DRIVE: DEFM 'Drive',0A0H
E5E3 65A0
                ;
E5E5 57726974 QUEST: DEFM 'Write to disk (y/n/c) ?',0A0H
E5E9 6520746F
E5ED 20646973
E5F1 6B202879
E5F5 2F6E2F63
E5F9 29203FA0
                ;
E5FD                   END
```

```
clear ,&he4af
Ok
bload "dskedt.bin",&he4b0
Ok
poke &he670,&hb0:poke &he671,&he4
Ok
mon

h]^e1,0,0,1

E9B9 21 00 84 AF 32 B4 EC 01 02 00 11 11 2F 3A 5D EF   !  ■ｯ2I●     /:]\
E9C9 B7 20 04 06 02 1E 1B AF 3C E5 D5 C5 CD 9A 36 C1   ｷ          ｯ<◣ｺﾅﾍ└6ﾁ
E9D9 D1 E1 30 08 3A B4 EC FE 03 20 EC C9 AF 32 B4 EC   ﾑ ﾎ0  :I●    ●ﾉｯ2I●
E9E9 15 28 0B 0C 24 7B B9 20 DE 04 0E 01 18 D9 CD 00    (   ${ｹ ﾞ      ﾙﾍ
E9F9 84 3E 30 D3 F3 21 FF 7F 2B 7C B5 20 FB 3E 00 D3   ■>0ﾓ月!   +¦ｵ   >  ﾓ
EA09 F3 C3 00 88 00 00 00 C3 23 71 23 77 EB 10 EE C9   月ﾃ  |     ﾃ#q#w♣ ／ﾉ
EA19 DC EC AC D9 C4 EC A2 D0 C1 EC DE D9 D0 EC 14 D4   ﾜ●ｬﾙﾄ●｢ﾐﾁ●ﾞﾙﾐ●  ﾔ
EA29 AE ED 8E D9 E8 EC A4 D4 B1 ED 4A CE EE EC EB D4   ｮ○■ﾙ♣●､ﾔｱ○Jﾎ／●♣ﾔ
```

```
EA39 F1 EC 6F D9 F4 EC 82 D9 6F ED FE CF AB ED E1 CE        円●oル日●▁ルoO マォO卜ホ
EA49 FF ED 76 CE BB EC 4B C3 D3 EC D8 ED D6 EC D8 ED         Oνホサ●Kテモ●リOヨ●リO
EA59 9F ED D8 ED A2 ED D8 ED A2 ED A5 ED A8 ED D8 ED        ノOリO「OリO「O・OィOリO
EA69 74 EE D8 ED 77 EE D8 ED 7A EE E0 DB 7D EE 22 C2        t／リOw／リOz／═□》／゛ツ
EA79 80 EE CC DA 83 EE F6 DA 86 EE A5 DE 89 EE 6D DB        ▁／フレ▬／ダレ■／・゛▌／m□
EA89 8C EE 56 D2 8F EE 96 CE 92 EE 8A D0 95 EE 08 D5        ▌／Vメ+／|ホ┤／▌ミー／ ユ
EA99 98 EE 36 D5 9B EE 49 D3 9E EE 44 D3 BC EE 00 C1         r／6ユ┘／Iモ╲／Dモシ／ チ
EAA9 BF EE 23 DA C2 EE E7 D9 C5 EE A2 DE C8 EE EC           ソ／#レツ／◤ルナ／「゛ネ／●

Drive 01,Track 00,Surface 00,Sector 01

Write to disk (y/n/c) ? c

h]^b
Ok
```

第3章　N88-Disk BASICの解析

——Analysis of N88-Disk BASIC Interpreter——

本章では，N_{88}－ＢＡＳＩＣ，N_{88}－Disk ＢＡＳＩＣインタプリタの，主にディスク，ファイルに関するシステムサブルーチンについて解説してゆきます．各サブルーチンはエントリ・アドレスの順にまとめることにします．

アドレス **各ルーチンのエントリ・アドレスです．**

機　能 **各ルーチンの機能を簡単に示します．**

レジスタ **各ルーチンの実行後，その内容の変化するレジスタを示します．もし，これらのレジスタの値が変化して不都合の生じる場合などは，その各ルーチンを呼び出す前にレジスタの待避が必要です．また，N_{88}－Disk ＢＡＳＩＣではインデックスレジスタ（ＩＸ，ＩＹ）及び裏レジスタを使用していませんので，それらについては省略しました．**

解　説 各ルーチンの機能を詳しく説明し，入出力パラメータ・注意事項なども示しました．

サンプル 各ルーチンを利用した簡単なサンプル・プログラムをアセンブルリストの形で付けておきます．プログラムは全てマシン語で記述されており，ＰＣ－８８０１の内蔵モニタより入力可能ですが，使用したアセンブラがザイログ仕様のため，内蔵アセンブラ（インテル仕様）では直接入力できません．従って，入力の際にはＡＰＰＥＮＤＩＸ（６章）のμＣＯＭ－８２・インテル・ニモニック対応表を利用するか，モニタのＥコマンドあるいはＳコマンドでオブジェクトコードを直接入力して下さい．

また実行例もリストの後に付けておきましたので御参照下さい．

３６９ＡＨ　ディスクと１セクタの入出力を行う.

アドレス	３６９ＡＨ
機　能	ディスクと１セクタの入出力を行なう.
レジスタ	Ａ，Ｆ，Ｄ，Ｅ，Ｈ，Ｌ

解　説　ディスクとの入出力を行なう最も基本的なサブルーチンです．パラメータは以下のとおりです.

Ｂレジスタ　………トラック・ナンバ.
Ｃレジスタ　………セクタ・ナンバ.
ＨＬレジスタ………データ・アドレス.
ＥＣ８５Ｈ　………ドライブ・ナンバ（０～）.
ＥＦ５ＤＨ　………ドライブ・タイプ.
ＣＹフラグ　………１：ライト，０：リード
Ｚフラグ　　………１：ベリファイ，０：リード

トラック・ナンバの指定には論理トラック・ナンバを用いますのでサーフェスの指定は必要ありません.

エラーの発生した時にはＣＹフラグをセットして戻ります.

サンプル

```
                ;
                ; --- disk read/write/verify
                ;
                ;       sample of subroutine from 369AH
                ;
                        ORG   0B900H
                ;
B900 2182B9     DISKRW:LD     HL,DRIVE          ; input drive number
B903 CD57B9            CALL   GETNUM
B906 D8                RET    C                 ; stop key return
                ;
B907 3D                DEC    A
B908 CD5899            CALL   9958H             ; set drive number and type
                ;
B90B 3AA7EC            LD     A,(0ECA7H)        ; check drive side
B90E B7                OR     A
B90F C46AB9            CALL   NZ,ISURF          ; double surface
                ;
B912 218DB9            LD     HL,TRACK          ; input track number
B915 CD57B9            CALL   GETNUM
B918 4F                LD     C,A
B919 3AA7EC            LD     A,(0ECA7H)
B91C B7                OR     A
B91D 79                LD     A,C
B91E 2802              JR     Z,DSKRW1
B920 07                RLCA                     ; track*2
B921 B0                OR     B                 ; add surface number
                ;
B922 47         DSKRW1:LD     B,A
```

```
B923 2196B9            LD   HL,SECTOR          ; input sector number
B926 CD57B9            CALL GETNUM
B929 4F                LD   C,A
                ;
B92A 21ABB9            LD   HL,DATAAD          ; input data address
B92D CD57B9            CALL GETNUM
                ;
B930 E5                PUSH HL
B931 C5                PUSH BC
B932 21BBB9     DSKRW2:LD   HL,MODE            ; input mode r/w/v
B935 CD72B9            CALL DSPMES
B938 CDC85F            CALL 5FC8H              ; line input subroutine
B93B 23                INC  HL
B93C 7E                LD   A,(HL)
B93D FE77              CP   'w'                ; write
B93F 37                SCF
B940 280B              JR   Z,DSKRW3           ; CY=1
                ;
B942 FE76              CP   'v'                ; verify
B944 2807              JR   Z,DSKRW3           ; CY=0,Z=1
                ;
B946 FE72              CP   'r'                ; read
B948 20E8              JR   NZ,DSKRW2
                ;
B94A 3E01              LD   A,1
B94C B7                OR   A                  ; CY=0,Z=0
                ;
B94D C1         DSKRW3:POP  BC
B94E E1                POP  HL
                ;
B94F CD9A36            CALL 369AH              ; access disk
                ;
B952 DC7BB9            CALL C,ERROR            ; CY=1 error
                ;
B955 18A9              JR   DISKRW
                ;
B957 C5         GETNUM:PUSH BC                 ; input number subroutine
B958 CD72B9            CALL DSPMES
B95B CDC85F            CALL 5FC8H
                ;
B95E F5                PUSH AF
B95F 23                INC  HL
B960 CDBC26            CALL 26BCH              ; convert to binary
B963 CDA021            CALL 21A0H              ; convert to integer
B966 F1                POP  AF
B967 7D                LD   A,L
B968 C1                POP  BC
B969 C9                RET
                ;
B96A 21A0B9     ISURF: LD   HL,SURFAC          ; input surface number
B96D CD57B9            CALL GETNUM
B970 47                LD   B,A
B971 C9                RET
                ;
B972 7E         DSPMES:LD   A,(HL)             ; display message subroutine
B973 A7                AND  A
B974 C8                RET  Z
B975 CD0D3E            CALL 3E0DH
B978 23                INC  HL
B979 18F7              JR   DSPMES
                ;
B97B 21DBB9     ERROR: LD   HL,ERRMES          ; error message output
B97E CD72B9            CALL DSPMES
B981 C9                RET
                ;
B982 0D0A4472 DRIVE: DEFM 0DH,0AH,'Drive ? ',0
B986 69766520
B98A 3F2000
B98D 54726163 TRACK: DEFM 'Track ? ',0
B991 6B203F20
B995 00
B996 53656374 SECTOR:DEFM 'Sector ? ',0
B99A 6F72203F
B99E 2000
B9A0 53757266 SURFAC:DEFM 'Surface ? ',0
B9A4 61636520
B9A8 3F2000
```

```
B9AB 44617461 DATAAD:DEFM 'Data address ? ',0
B9AF 20616464
B9B3 72657373
B9B7 203F2000
                 ;
B9BB 0D0A5772 MODE:  DEFM 0DH,0AH,'Write(w)/Read(r)'
B9BF 69746528
B9C3 77292F52
B9C7 65616428
B9CB 7229
B9CD 2F566572        DEFM '/Verify(v) ? ',0
B9D1 69667928
B9D5 7629203F
B9D9 2000
                 ;
B9DB 4469736B ERRMES:DEFM 'Disk I/O error !!',7,0
B9DF 20492F4F
B9E3 20657272
B9E7 6F722021
B9EB 210700
                 ;
B9EE                 END
```

```
a=&hb900:call a

Drive ? 1
Surface ? 1
Track ? 18
Sector ? 1
Data address ? &hc000

Write(w)/Read(r)/Verify(v) ? r

Drive ?
Ok
mon

h]dc000,c0ff
C000 62 61 63 6B 75 70 6E 38 38 90 48 FF FF FF FF FF        backupn88┴H
C010 66 69 6C 65 20 20 64 61 74 00 49 FF FF FF FF FF        file  dat I
C020 73 65 74 69 6E 66 6E 38 38 80 45 FF FF FF FF FF        setinfn88_E
C030 78 66 69 6C 65 73 6E 38 38 80 44 FF FF FF FF FF        xfilesn88_D
C040 73 79 73 67 65 6E 20 20 20 80 4C FF FF FF FF FF        sysgen    ⊥
C050 64 75 6D 6D 79 31 20 20 20 40 46 FF FF FF FF FF        dummy1    @F
C060 64 75 6D 6D 79 32 20 20 20 40 43 FF FF FF FF FF        dummy2    @C
C070 64 75 6D 6D 79 33 20 20 20 40 4D FF FF FF FF FF        dummy3    @M
C080 73 61 6D 70 6C 65 64 61 74 40 42 FF FF FF FF FF        sampledat@B
C090 64 75 6D 6D 79 34 20 20 20 40 4E FF FF FF FF FF        dummy4    @N
C0A0 6D 61 63 68 69 6E 65 20 20 41 41 FF FF FF FF FF        machine   AA
C0B0 73 61 6D 70 6C 65 62 69 6E 41 4F FF FF FF FF FF        samplebinAO
C0C0 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
C0D0 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
C0E0 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
C0F0 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
h]^b
Ok
```

３６Ｅ２Ｈ　ＰＣ－８０３１のイニシァライズ．

アドレス　３６Ｅ２Ｈ

機　　能　ＰＣ－８０３１のイニシァライズを行う．

レジスタ　Ａ，Ｆ，Ｄ

解　　説　ＰＣ－８０３１，ＰＣ－８０３１－２Ｗ（または同等のインテリジェントタイプのディスクユニット）とのインタフェース（μＰＤ８２５５）をイニシァライズし，ディスクユニットに対しイニシァライズコマンドを送出した後，接続されているミニディスクユニットの数を求めアキュムレータに格納して戻ります．

ディスクユニットが接続されていない場合にはアキュムレータには００Ｈが与えられます．

サンプル

```
               ;
               ; --- initialize disk unit
               ;
               ;       sample of subroutine from 36E2H
               ;
                       ORG  0B900H
               ;
B900 CDE236    INIT:   CALL 36E2H              ; init disk unit
               ;
B903 F5                PUSH AF
B904 2120B9            LD   HL,MESS1           ; initialize message
B907 CD17B9            CALL DSPMES
B90A F1                POP  AF
               ;
B90B C630              ADD  A,'0'              ; convert to ascii
B90D CD0D3E            CALL 3E0DH
B910 2142B9            LD   HL,MESS2           ; drive message
B913 CD17B9            CALL DSPMES
               ;
B916 C9                RET
               ;
B917 7E        DSPMES:LD    A,(HL)             ; display message
B918 A7                AND  A
B919 C8                RET  Z
B91A CD0D3E            CALL 3E0DH
B91D 23                INC  HL
B91E 18F7              JR   DSPMES
               ;
B920 0D0A4469  MESS1:  DEFM 0DH,0AH,'Disk unit is '
B924 736B2075
B928 6E697420
B92C 697320
B92F 696E6974          DEFM 'initialized and ',0DH,0AH,0
B933 69616C69
B937 7A656420
B93B 616E6420
B93F 0D0A00
               ;
B942 20647269  MESS2:  DEFM ' drives are connected.',0
B946 76657320
```

```
B94A 61726520
B94E 636F6E6E
B952 65637465
B956 642E00
                    ;
B959                          END
```

```
a=&hb900:call a

Disk unit is initialized and
2 drives are connected.
Ok
```

３７２２H　PC－８０３１にコマンド・パラメータを送信する.

アドレス　３７２２H

機　　能　PC－８０３１にコマンド・パラメータを送信する.

レジスタ　A，F

解　　説　PC－８０３１，PC－８０３１－２W（または同等のインテリジェントタイプのディスクユニット）へコマンド，パラメータを送信するサブルーチンで，コマンド・ナンバ，ドライブ・ナンバなどの指定は以下の様に行います.

Aレジスタ………コマンド・ナンバ.
Bレジスタ………トラック・ナンバ.
Cレジスタ………セクタ・ナンバ.
EF４AH………セクタ数.
EF５FH………PC－８０３１，PC－８０３１－２W内におけるドライブ・ナンバ（０～）.

このサブルーチンが利用できるのは，パラメータの送信フォーマットが以下の様になっているものに限ります.

コマンド・ナンバ.
セクタ数.
ドライブ・ナンバ（０～）.
トラック・ナンバ.
セクタ・ナンバ.

これらに該当するコマンドについては，５－２を御参照下さい.

サンプル

```
                    ;
                    ; --- write PC-8031's farmware to disk
                    ;
                    ;       sample of subroutine from 3722H
                    ;
                            ORG   0B900H
                    ;
B900 2132B9                 LD    HL,MESS           ; message for loading
B903 CD29B9                 CALL  DSPMES
                    ;
B906 3E10                   LD    A,16              ; number of sector
B908 324AEF                 LD    (0EF4AH),A
B90B 3E01                   LD    A,1               ; drive number
B90D 325FEF                 LD    (0EF5FH),A
                    ;
B910 3E0F                   LD    A,15              ; save data command
B912 010100                 LD    BC,0001H          ; track 00,sector 01
                    ;
B915 CD2237                 CALL  3722H             ; send parameters
```

```
                   ;
B918 3E00                LD   A,0                ; address 0000H
B91A CDD237              CALL 37D2H              ; send data subroutine
B91D 3E00                LD   A,0
B91F CDD237              CALL 37D2H
                   ;
B922 214BB9              LD   HL,COMP            ; message for complete
B925 CD29B9              CALL DSPMES
                   ;
B928 C9                  RET
                   ;
B929 7E            DSPMES:LD  A,(HL)             ; display message subroutine
B92A A7                  AND  A
B92B C8                  RET  Z
B92C CD0D3E              CALL 3E0DH
B92F 23                  INC  HL
B930 18F7                JR   DSPMES
                   ;
B932 0D0A4E6F MESS:  DEFM 0DH,0AH,'Now saving farmware...',0
B936 77207361
B93A 76696E67
B93E 20666172
B942 6D776172
B946 652E2E2E
B94A 00
                   ;
B94B 0D0A5361 COMP:  DEFM 0DH,0AH,'Save complete.',0
B94F 76652063
B953 6F6D706C
B957 6574652E
B95B 00
                   ;
B95C                     END
```

```
a=&hb900:call a

Now saving farmware...
Save complete.
Ok
mon

h]^r4,0,0,1,c000,c0ff
h]lc000
C000    C3 007E      JMP  007E
C003    00           NOP
C004    00           NOP
C005    00           NOP
C006    00           NOP
C007    00           NOP
C008    F3           DI
C009    CD 7F22      CALL 7F22
C00C    C3 004E      JMP  004E
C00F    00           NOP
h]^b
Ok
```

373AH　PC－8031のリードバッファからデータを読み込む．

アドレス　373AH

機　　能　PC－8031のリードバッファからデータを読み込む．

レジスタ　A，F，B，C，D，E，H，L

解　　説　PC－8031，PC－8031－2W（または同等のインテリジェントタイプのディスクドライブ）のリードバッファより連続したデータ列を受け取り，メモリに格納するサブルーチンで，データ列の長さなどのパラメータの指定は以下の様に行います．

　　Bレジスタ　………転送バイト数／256の値．
　　DEレジスタ………データを格納する領域の先頭アドレス．
　　EF5DH　………ドライブ・タイプ（2，3）．

　このサブルーチンでは，ドライブが片面仕様ならば1バイトずつ，両面仕様ならば2バイトずつデータを転送します．また，指定するデータの長さは直前にコマンドナンバー2または22で，リードバッファに読み出したデータの長さと一致する必要があります．

サンプル

```
                ;
                ; --- read track 0
                ;
                ;       sample of subroutine from 373AH
                ;
                        ORG  0B900H
                ;
B900 2147B9             LD   HL,MESS            ; message for loading
B903 CD3EB9             CALL DSPMES
                ;
B906 3E00               LD   A,0                ; drive 1
B908 3285EC             LD   (0EC85H),A
B90B CDCB3D             CALL 3DCBH              ; get type
B90E 325DEF             LD   (0EF5DH),A
                ;
B911 3E02               LD   A,2                ; read data command
B913 CDC937             CALL 37C9H              ; send command subroutine
B916 3E10               LD   A,16               ; number of sector
B918 CDD237             CALL 37D2H              ; send data subroutine
B91B 3E00               LD   A,0                ; drive 1
B91D CDD237             CALL 37D2H
B920 3E00               LD   A,0                ; track 0
B922 CDD237             CALL 37D2H
B925 3E01               LD   A,1                ; sector 1
B927 CDD237             CALL 37D2H
                ;
B92A 3E03               LD   A,3                ; send data command
B92C CDC937             CALL 37C9H
                ;
B92F 1100C0             LD   DE,0C000H          ; top of buffer
B932 0610               LD   B,16
```

```
                  ;
B934 CD3A37              CALL 373AH             ; load to buffer
                  ;
B937 2160B9              LD   HL,COMP
B93A CD3EB9              CALL DSPMES
                  ;
B93D C9                  RET
                  ;
B93E 7E           DSPMES:LD   A,(HL)            ; display message subroutine
B93F A7                  AND  A
B940 C8                  RET  Z
B941 CD0D3E              CALL 3E0DH
B944 23                  INC  HL
B945 18F7                JR   DSPMES
                  ;
B947 0D0A4E6F MESS:  DEFM 0DH,0AH,'Now loading track 0...',0
B94B 77206C6F
B94F 6164696E
B953 67207472
B957 61636B20
B95B 302E2E2E
B95F 00
                  ;
B960 0D0A4C6F COMP:  DEFM 0DH,0AH,'Load complete.',0
B964 61642063
B968 6F6D706C
B96C 6574652E
B970 00
                  ;
B971                     END
```

```
a=&hb900:call a

Now loading track 0...
Load complete.
Ok
mon

h]lc000
C000    21 8400     LXI   H,8400
C003    AF          XRA   A
C004    32 ECB4     STA   ECB4
C007    01 0002     LXI   B,0002
C00A    11 2F11     LXI   D,2F11
C00D    3A EF5D     LDA   EF5D
h]^b
Ok
```

３７６６Ｈ　ＰＣ－８０３１へ連続したデータを送信する．

アドレス	３７６６Ｈ
機　能	ＰＣ－８０３１へ連続したデータを送信する．
レジスタ	Ａ，Ｆ，Ｂ，Ｃ，Ｄ，Ｅ，Ｈ，Ｌ

解　説　ＰＣ－８０３１，ＰＣ－８０３１－２Ｗ（または同等のインテリジェントタイプのディスクユニット）へ連続したデータ列を送信するサブルーチンで，データの長さなどのパラメータの指定は以下の様に行います．

　　Ｂレジスタ　………転送バイト数／２５６の値．
　　ＤＥレジスタ………データの格納されている領域の先頭アドレス．
　　ＥＦ５ＤＨ　………ドライブ・タイプ（２，３）．

このサブルーチンでは，ドライブが片面仕様ならば１バイトづつ，両面仕様ならば２バイトづつデータを送信します．したがって，ドライブが両面仕様の場合にはＰＣ－８０３１－２Ｗに対し高送転送用のコマンド（５－２を参照）が実行されており，かつＥＦ１５Ｈ番地に００Ｈ以外の値が書き込まれている必要があります．

サンプル

```
                ;
                ; --- save memory data to disk
                ;
                ;       sample of subroutine from 3766H
                ;
                        ORG  0B900H
                ;
B900 3E01               LD   A,1             ; drive 2
B902 3285EC             LD   (0EC85H),A
B905 CDCB3D             CALL 3DCBH           ; get drive type
B908 325DEF             LD   (0EF5DH),A
                ;
B90B 217BB9             LD   HL,TOPADR       ; input top address
B90E CD69B9             CALL GETNUM
                ;
B911 EB                 EX   DE,HL
B912 218CB9             LD   HL,ENDADR       ; input end address
B915 CD69B9             CALL GETNUM
                ;
B918 EB                 EX   DE,HL
B919 E5                 PUSH HL
B91A CD2F23             CALL 232FH           ; get bytes
B91D D1                 POP  DE
B91E 24                 INC  H
B91F 2E00               LD   L,0
                ;
B921 010100             LD   BC,0001H        ; top of track,sector
                ;
B924 CD45B9     WRTMEL:CALL  SENDPR          ; send parameter
                ;
B927 C5                 PUSH BC
```

```
B928 E5             PUSH HL
B929 0601           LD   B,1
B92B CD6637         CALL 3766H          ; send data
B92E E1             POP  HL
B92F C1             POP  BC
               ;
B930 25             DEC  H
B931 280B           JR   Z,EXIT
               ;
B933 0C             INC  C              ; sector increment
B934 79             LD   A,C
B935 FE11           CP   17
B937 38EB           JR   C,WRTMEL
B939 0E01           LD   C,1
B93B 04             INC  B
B93C 18E6           JR   WRTMEL
               ;
B93E 219BB9    EXIT: LD  HL,COMP
B941 CD60B9          CALL DSPMES
B944 C9              RET
               ;
B945 3E11      SENDPR:LD  A,17          ; write disk command
B947 3215EF         LD   (0EF15H),A     ; set fast flag
B94A CDCC37         CALL 37CCH          ; send command
B94D 3E01           LD   A,1            ; number of sector
B94F CDD237         CALL 37D2H          ; send data
B952 3E01           LD   A,1            ; drive 2
B954 CDD237         CALL 37D2H
B957 78             LD   A,B            ; track
B958 CDD237         CALL 37D2H
B95B 79             LD   A,C            ; sector
B95C CDD237         CALL 37D2H
               ;
B95F C9             RET
               ;
B960 7E        DSPMES:LD  A,(HL)        ; display message subroutine
B961 A7             AND  A
B962 C8             RET  Z
B963 CD0D3E         CALL 3E0DH
B966 23             INC  HL
B967 18F7           JR   DSPMES
               ;
B969 C5        GETNUM:PUSH BC           ; get number subroutine
B96A D5             PUSH DE
B96B CD60B9         CALL DSPMES
B96E CDC85F         CALL 5FC8H          ; line input from keyboard
               ;
B971 23             INC  HL
B972 CDBC26         CALL 26BCH          ; convert to binary
B975 CDA021         CALL 21A0H          ; convert to integer
B978 D1             POP  DE
B979 C1             POP  BC
B97A C9             RET
               ;
B97B 0D0A546F  TOPADR:DEFM 0DH,0AH,'Top address ? ',0
B97F 70206164
B983 64726573
B987 73203F20
B98B 00
B98C 456E6420  ENDADR:DEFM 'End address ? ',0
B990 61646472
B994 65737320
B998 3F2000
               ;
B99B 53617665  COMP: DEFM 'Save complete.',0
B99F 20636F6D
B9A3 706C6574
B9A7 652E00
               ;
B9AA                END
```

```
a=&hb900:call a
```

```
Top address ? 0
End address ? &h7fff
Save complete.
Ok
mon

h]^r4,0,0,1,c000,c0ff
h]lc000
C000   F3           DI
C001   31 E1A0      LXI   SP,E1A0
C004   C3 77F7      JMP   77F7
C007   00           NOP
C008   7E           MOV   A,M
C009   E3           XTHL
C00A   BE           CMP   M
C00B   23           INX   H
C00C   E3           XTHL
C00D   C2 0393      JNZ   0393
h]^b
Ok
```

３７９０Ｈ　コマンドの実行結果を受信する．

アドレス　３７９０Ｈ

機　　能　ＰＣ－８０３１に対するコマンドの実行結果を受信する．

レジスタ　Ａ，Ｆ

解　　説　ＰＣ－８０３１，ＰＣ－８０３１－２Ｗ（または同等のインテリジェントタイプのディスクユニット）に対して，直前に送ったコマンドの実行結果を問い合わせるサブルーチンです．エラーの発生のないときは，このサブルーチンからはＲＥＴ命令により戻りますが，エラーの発生の認められるときには，スタックポインタ（ＳＰ）を４レベル戻し（ＳＰ←ＳＰ＋８），ＣＹフラグをセットして戻ります．また，このときＥＣＢ４Ｈ番地の値をインクリメントさせていき，ＥＣＢ４Ｈ番地の値が４に達した（エラーが４回）場合には，ＥＣＢ６Ｈ番地の値を調べ，００Ｈ以外の値ならばDISK Ｉ／Ｏ errorを発生させます．ＥＣＢ６Ｈ番地の値が００Ｈならば，スタックポインタ（ＳＰ）をさらに６レベル戻し（ＳＰ←ＳＰ＋１２），ＣＹフラグをセットして戻ります．

なお，このサブルーチンは単独で使用する場合，上記の様にスタックの不一致が生ずることがありますので，使用の際はこの点に御注意下さい．

サンプル

```
                ;
                ; --- check all track
                ;
                ;       sample of subroutine from 3790H
                ;
                        ORG  0B900H
                ;
B900 0600       CHKALT:LD   B,0              ; top of track
                ;
B902 C5         CHKL:   PUSH BC
B903 AF                 XOR  A               ; clear error counter
B904 32B4EC             LD   (0ECB4H),A
                ;
B907 3E02               LD   A,2             ; read to buffer command
B909 CDC937             CALL 37C9H           ; send command subroutine
B90C 3E10               LD   A,16            ; all sectors
B90E CDD237             CALL 37D2H           ; send data subroutine
B911 3E00               LD   A,0             ; drive 1
B913 CDD237             CALL 37D2H
B916 78                 LD   A,B             ; track
B917 CDD237             CALL 37D2H
B91A 3E01               LD   A,1             ; sector
B91C CDD237             CALL 37D2H
                ;
B91F 2129B9             LD   HL,RETADR       ; set return address to stack
B922 E5                 PUSH HL
B923 C5                 PUSH BC
B924 D5                 PUSH DE
B925 F5                 PUSH AF
                ;
B926 CD9037             CALL 3790H           ; check result of command
```

```
                   ;
B929 3AB4EC        RETADR:LD     A,(0ECB4H)          ; check error
B92C B7                   OR     A
B92D 2004                 JR     NZ,CHKE             ; abnormal end
                   ;
B92F F1                   POP    AF
B930 D1                   POP    DE
B931 C1                   POP    BC
B932 E1                   POP    HL
                   ;
B933 3AB4EC        CHKE:  LD     A,(0ECB4H)
B936 B7                   OR     A
B937 C442B9               CALL   NZ,ERROR
                   ;
B93A C1                   POP    BC
B93B 04                   INC    B
B93C 78                   LD     A,B
B93D FE50                 CP     80                  ; last track
B93F 38C1                 JR     C,CHKL
                   ;
B941 C9                   RET
                   ;
B942 2180B9        ERROR: LD     HL,ERRMES           ; message for error
B945 CD64B9               CALL   DSPMES
                   ;
B948 D1                   POP    DE
B949 C1                   POP    BC
B94A C5                   PUSH   BC
B94B D5                   PUSH   DE
B94C 68                   LD     L,B
B94D 2600                 LD     H,0
B94F CDC228               CALL   28C2H               ; output integer
                   ;
B952 216DB9               LD     HL,ASK              ; ask abort ?
B955 CD64B9               CALL   DSPMES
B958 CDC85F               CALL   5FC8H               ; line input from keyboard
B95B 23                   INC    HL
B95C 7E                   LD     A,(HL)
B95D FE79                 CP     'y'
B95F 2002                 JR     NZ,ERRORE
                   ;
B961 F1                   POP    AF                  ; return cancel
B962 C1                   POP    BC
                   ;
B963 C9            ERRORE:RET
                   ;
B964 7E            DSPMES:LD     A,(HL)              ; display message subroutine
B965 A7                   AND    A
B966 C8                   RET    Z
B967 CD0D3E               CALL   3E0DH
B96A 23                   INC    HL
B96B 18F7                 JR     DSPMES
                   ;
B96D 0D0A4162      ASK:   DEFM   0DH,0AH,'Abort(y/else) ? ',0
B971 6F727428
B975 792F656C
B979 73652920
B97D 3F2000
                   ;
B980 4572726F      ERRMES:DEFM   'Error in track ',7,0
B984 7220696E
B988 20747261
B98C 636B2007
B990 00
                   ;
B991                      END
```

```
a=&hb900:call a
Error in track 1
Abort(y/else) ? n
Error in track 76
Abort(y/else) ? n
```

```
Error in track 77
Abort(y/else) ? n
Error in track 78
Abort(y/else) ? n
Error in track 79
Abort(y/else) ? n
Ok
```

３７Ｃ９Ｈ　ＰＣ－８０３１へコマンドを送出する．

アドレス	３７Ｃ９Ｈ
機　能	ＰＣ－８０３１へコマンドを送出する．
レジスタ	Ａ，Ｆ

解　説　ＰＣ－８０３１，ＰＣ－８０３１－２Ｗ（または同等のインテリジェントタイプのディスクドライブ）にアキュムレータの値をコマンドとして送出します．エラーの発生した場合にはＣＹフラグをセットして戻ります．

このサブルーチンではコマンドナンバ１，３に関して，それぞれコマンドナンバ１７，１８への変換を行ないます(５－２を参照)．また，それらのコマンドに対しては，同時に高速転送フラグ（ＥＦ１５Ｈ）のセットが行なわれますが，他のコマンドに対しては，フラグはリセットされます．フラグをコマンドに応じてセットしたい場合には３７ＣＣＨ番地にエントリするのが有効です．

なお，このサブルーチンはデータの送出時にＡＴＮビットをセットする以外は３７Ｄ２Ｈからのサブルーチンと共通です．

サンプル

```
                ;
                ; --- physical format a disk
                ;
                ;       sample of subroutine from 37C9H
                ;
                        ORG   0B900H
                ;
B900 2175B9     FORMAT:LD    HL,TITLE            ; title display
B903 CD4BB9             CALL  DSPMES
                ;
B906 2198B9     FORML: LD    HL,DRIVE            ; drive number input
B909 CD4BB9             CALL  DSPMES
B90C CDC85F             CALL  5FC8H              ; subroutine line input
B90F D8                 RET   C                  ; stop key return
                ;
B910 23                 INC   HL
B911 CDBC26             CALL  26BCH              ; convert to binary
B914 CDA021             CALL  21A0H              ; convert to integer
B917 7D                 LD    A,L                ; lower byte of FAC
                ;
B918 F5                 PUSH  AF
B919 CD5399             CALL  9953H              ; set disk identification
                ;
B91C 3E05               LD    A,5                ; format command
B91E CDC937             CALL  37C9H              ; send command
                ;
B921 F1                 POP   AF
B922 3D                 DEC   A
B923 CDD237             CALL  37D2H              ; send data
                ;
B926 21BCB9             LD    HL,FMTMES          ; message in format
B929 CD4BB9             CALL  DSPMES
                ;
```

```
B92C CD54B9            CALL DELAY            ; wait about 40 seconds
                  ;
B92F 3E0F              LD   A,0FH            ; ATN on
B931 D3FF              OUT  (0FFH),A
                  ;
B933 DBFE       WAIT:  IN   A,(0FEH)         ; input control bit
B935 E602              AND  2                ; check RFD bit
B937 28FA              JR   Z,WAIT
                  ;
B939 3E06              LD   A,6              ; result status command
B93B CDC937            CALL 37C9H
                  ;
B93E CD4738            CALL 3847H            ; recieve data
B941 E601              AND  1                ; check error bit
B943 C465B9            CALL NZ,ERROR
B946 CC6EB9            CALL Z,COMP
                  ;
B949 18BB              JR   FORML
                  ;
B94B 7E         DSPMES:LD   A,(HL)
B94C A7                AND  A
B94D C8                RET  Z
B94E CD0D3E            CALL 3E0DH
B951 23                INC  HL
B952 18F7              JR   DSPMES
                  ;
B954 212600     DELAY: LD   HL,38            ; delay for 40 seconds
B957 110000     DELL1: LD   DE,0
B95A 1B         DELL2: DEC  DE
B95B 7B                LD   A,E
B95C B2                OR   D
B95D 20FB              JR   NZ,DELL2
B95F 2B                DEC  HL
B960 7D                LD   A,L
B961 B4                OR   H
B962 20F3              JR   NZ,DELL1
B964 C9                RET
                  ;
B965 F5         ERROR: PUSH AF               ; error message
B966 21A3B9            LD   HL,ERRMES
B969 CD4BB9            CALL DSPMES
B96C F1                POP  AF
B96D C9                RET
                  ;
B96E 21AFB9     COMP:  LD   HL,CMPMES        ; message for complete
B971 CD4BB9            CALL DSPMES
B974 C9                RET
                  ;
B975 0D0A2D2D TITLE: DEFM 0DH,0AH,'--- PHYSICAL FORMAT
B979 2D205048
B97D 59534943
B981 414C2046
B985 4F524D41
B989 5420
B98B 41204449          DEFM 'A DISK ---',0DH,0AH,0
B98F 534B202D
B993 2D2D0D0A
B997 00
                  ;
B998 0D0A4472 DRIVE: DEFM 0DH,0AH,'Drive ? ',0
B99C 69766520
B9A0 3F2000
                  ;
B9A3 4572726F ERRMES:DEFM 'Error !!',7,0DH,0AH,0
B9A7 72202121
B9AB 070D0A00
                  ;
B9AF 436F6D70 CMPMES:DEFM 'Completed.',0DH,0AH,0
B9B3 6C657465
B9B7 642E0D0A
B9BB 00
                  ;
B9BC 466F726D FMTMES:DEFM 'Formatting...',0DH,0AH,0
```

```
B9C0 61747469
B9C4 6E672E2E
B9C8 2E0D0A00
                   ;
B9CC                         END
```

```
a=&hb900:call a

--- PHYSICAL FORMAT A DISK ---

Drive ? 1
Formatting...
Completed.

Drive ?
Ok
```

３７Ｄ２Ｈ　ＰＣ－８０３１へデータを送出する．

アドレス	３７Ｄ２Ｈ
機　能	ＰＣ－８０３１へデータを送出する．
レジスタ	Ｆ

解　説 ＰＣ－８０３１，ＰＣ－８０３１－２Ｗ（または同等のインテリジェントタイプのディスクドライブ）にアキュムレータの値をデータとして送出します．また，ＥＦ１５Ｈ番地に００Ｈ以外の値がセットされていて，かつ高速転送用のコマンドの実行中にはＬレジスタのデータも送出し，１度のサブルーチンコールで２バイト送ることができます（ただしＰＣ－８０３１－２Ｗのみ）．

エラーの発生した場合にはＣＹフラグをセットして戻ります．

サンプル

```
                ;
                ; --- copy all track
                ;
                ;        sample of subroutine from 37D2H
                ;
                        ORG   0B900H
                ;
B900 2165B9     COPY:   LD    HL,TITLE           ; display title
B903 CD47B9             CALL  DSPMES
                ;
B906 2180B9             LD    HL,MESS1           ; check sure
B909 CD47B9             CALL  DSPMES
                ;
B90C CD8335             CALL  3583H              ; wait 1 key
                ;
B90F 1600               LD    D,0                ; top of track
                ;
B911 CD50B9     COPYL:  CALL  DSPTRK
                ;
B914 3E04               LD    A,4                ; copy command
B916 CDC937             CALL  37C9H              ; send command
B919 3E10               LD    A,16               ; sector/track
B91B CDD237             CALL  37D2H              ; send data
                ;
B91E 3E00               LD    A,0                ; source drive
B920 CDD237             CALL  37D2H
B923 7A                 LD    A,D                ; track
B924 CDD237             CALL  37D2H
B927 3E01               LD    A,1                ; top of sector
B929 CDD237             CALL  37D2H
                ;
B92C 3E01               LD    A,1                ; distination drive
B92E CDD237             CALL  37D2H
B931 7A                 LD    A,D
B932 CDD237             CALL  37D2H
B935 3E01               LD    A,1
B937 CDD237             CALL  37D2H
                ;
B93A 14                 INC   D
B93B 7A                 LD    A,D
B93C FE50               CP    80                 ; all track copied ?
B93E 20D1               JR    NZ,COPYL
```

```
                    ;
B940 21E3B9                LD   HL,COMP
B943 CD47B9                CALL DSPMES
                    ;
B946 C9                    RET
                    ;
B947 7E             DSPMES:LD   A,(HL)
B948 A7                    AND  A
B949 C8                    RET  Z
B94A CD0D3E                CALL 3E0DH
B94D 23                    INC  HL
B94E 18F7                  JR   DSPMES
                    ;
B950 D5             DSPTRK:PUSH DE
B951 21D4B9                LD   HL,TRACK             ; track number display
B954 CD47B9                CALL DSPMES
                    ;
B957 6A                    LD   L,D
B958 2600                  LD   H,0
B95A CDC228                CALL 28C2H
B95D 21D1B9                LD   HL,CRLF
B960 CD47B9                CALL DSPMES
B963 D1                    POP  DE
B964 C9                    RET
                    ;
B965 0D0A2D2D TITLE: DEFM 0DH,0AH,'--- COPY ALL TRACK ---',0DH,0AH,0
B969 2D20434F
B96D 50592041
B971 4C4C2054
B975 5241434B
B979 202D2D2D
B97D 0D0A00
                    ;
B980 53657420 MESS1: DEFM 'Set source disk to drive 1',0DH,0AH
B984 736F7572
B988 63652064
B98C 69736B20
B990 746F2064
B994 72697665
B998 20310D0A
B99C 53657420        DEFM 'Set formatted disk to '
B9A0 666F726D
B9A4 61747465
B9A8 64206469
B9AC 736B2074
B9B0 6F20
B9B2 64726976        DEFM 'drive 2',0DH,0AH
B9B6 6520320D
B9BA 0A
B9BB 53757265        DEFM 'Sure(y:hit any key)?',0DH,0AH
B9BF 28793A68
B9C3 69742061
B9C7 6E79206B
B9CB 6579293F
B9CF 0D0A
                    ;
B9D1 0D0A00    CRLF:  DEFM 0DH,0AH,0
                    ;
B9D4 436F7079 TRACK: DEFM 'Copying track ',0
B9D8 696E6720
B9DC 74726163
B9E0 6B2000
                    ;
B9E3 436F7079 COMP:  DEFM 'Copy completed.',0
B9E7 20636F6D
B9EB 706C6574
B9EF 65642E00

B9F3                 END


a=&hb900:call a

--- COPY ALL TRACK ---
```

```
Set source disk to drive 1
Set formatted disk to drive 2
Sure(y:hit any key)?

Copying track 0
Copying track 1


Copying track 78
Copying track 79
Copy completed.
Ok
```

３８４７H　PC－８０３１よりデータを受信する.

アドレス	３８４７H
機　　能	PC－８０３１よりデータを受信する.
レジスタ	A, F, L

解　　説　PC－８０３１, PC－８０３１－２W（または同等のインテリジェントタイプのディスクドライブ）からデータを受け取り, 値をアキュムレータに格納して戻ります. また, EF１５H番地に００H以外の値がセットされていて, かつ高速転送用のコマンドの実行中にはさらにもう１バイト受け取り, 値をLレジスタに格納して戻るため, １度のサブルーチンコールで２バイト受け取ることができます（ただしPC－８０３１－２Wのみ).

エラーの発生した場合にはCYフラグをセットして戻ります.

サンプル

```
                ;
                ; --- check protect-notch
                ;
                ;       sample of subroutine from 3847H
                ;
                        ORG  0B900H
                ;
B900 2150B9     CHKPRO:LD   HL,DRIVE            ; drive number input
B903 CD40B9             CALL DSPMES
B906 CDC85F             CALL 5FC8H              ; subroutine line input
B909 D8                 RET  C
                ;
B90A 23                 INC  HL
B90B CDBC26             CALL 26BCH              ; convert to binary
B90E CDA021             CALL 21A0H              ; convert to integer
B911 7D                 LD   A,L                ; lower byte of FAC
                ;
B912 F5                 PUSH AF
B913 215BB9             LD   HL,MESS1
B916 CD40B9             CALL DSPMES
B919 F1                 POP  AF
B91A F5                 PUSH AF
B91B C630               ADD  A,'0'              ; convert to ascii
B91D CD0D3E             CALL 3E0DH
B920 2164B9             LD   HL,MESS2
B923 CD40B9             CALL DSPMES
                ;
B926 3E14               LD   A,20               ; device status command
B928 CDC937             CALL 37C9H
B92B F1                 POP  AF
                ;
B92C 3D                 DEC  A
B92D CDD237             CALL 37D2H              ; send device number
B930 CD4738             CALL 3847H              ; get device status
B933 CB77               BIT  6,A                ; check WP byte of ST3
B935 CC49B9             CALL Z,DSPNOT
                ;
B938 2169B9             LD   HL,MESS3
B93B CD40B9             CALL DSPMES
                ;
```

```
B93E 18C0             JR    CHKPRO
                ;
B940 7E         DSPMES:LD   A,(HL)
B941 A7               AND   A
B942 C8               RET   Z
B943 CD0D3E           CALL  3E0DH
B946 23               INC   HL
B947 18F7             JR    DSPMES
                ;
B949 217EB9     DSPNOT:LD   HL,NOTMES
B94C CD40B9           CALL  DSPMES
B94F C9               RET
                ;
B950 0D0A4472 DRIVE: DEFM 0DH,0AH,'Drive ? ',0
B954 69766520
B958 3F2000
                ;
B95B 0D0A4472 MESS1: DEFM 0DH,0AH,'Drive ',0
B95F 69766520
B963 00
B964 20697320 MESS2: DEFM ' is ',0
B968 00
B969 77726974 MESS3: DEFM 'write-protected !!',0DH,0AH,0
B96D 652D7072
B971 6F746563
B975 74656420
B979 21210D0A
B97D 00
                ;
B97E 6E6F7420 NOTMES:DEFM 'not ',0
B982 00
                ;
B983                  END
```

```
a=&hb900:call a

Drive ? 1

Drive 1 is write-protected !!

Drive ? 2

Drive 2 is not write-protected !!

Drive ?
Ok
```

３９７５Ｈ　ＤＭＡ・ＦＤＤとのデータ転送を行う.

アドレス　３９７５Ｈ

機　能　ＤＭＡ・ＦＤＤとのデータ転送を行う.

レジスタ　Ａ，Ｆ，Ｂ，Ｃ，Ｄ，Ｅ，Ｈ，Ｌ

解　説　ＤＭＡ転送方式のディスクドライブ（ＰＣ－８８８１など）とのデータ転送を行うサブルーチンで，読み書きの指定などのパラメータは以下の様になります.

ＥＦ１７Ｈ…………………モードの指定（０：イニシァライズ，２：ライト，３：リード）.
ＥＦ１８Ｈ…………………ＨＤ，ＵＳ１，ＵＳ０合成バイト（第２ビット：ヘッド・ナンバ，第１，第０ビット：ドライブ・ナンバ（０～３））.
ＥＦ１９Ｈ…………………シリンダ・ナンバ.
ＥＦ１ＡＨ…………………セクタ・ナンバ.
ＥＦ１ＢＨ…………………転送セクタ数.
ＥＦ１ＣＨ，１ＤＨ………データ・アドレス.

この他にも，８インチ用・５インチ用を区別するためのパラメータがＥＦ４ＢＨ～ＥＦ５ＣＨ番地に必要ですが，それらは全てまとめて３Ｄ６７Ｈ番地からのサブルーチンでセットすることができます.

また，エラーの発生した場合にはＣＹフラグをセットして戻りますが，その際きＥＦ３ＤＨにはエラーの情報が格納されています．それらは以下の様になっています.

８０Ｈ………ドライブが非レディ状態にあったことを示します.
８１Ｈ………引き渡したパラメータが範囲外であったことを示します.
８６Ｈ………書き込みにおいてエラーの発生したことを示します.
８７Ｈ………読み出しにおいてエラーの発生したことを示します.

サンプル

```
                 ;
                 ; --- backup DMA disk
                 ;
                 ;       sample of subroutine from 3975H
                 ;
                        ORG   0C000H
                 ;
C000 2183C0      BACKUP:LD    HL,PROMPT          ; print prompt
C003 CD7AC0             CALL  DSPMES
C006 CD8335             CALL  3583H
                 ;
```

```
C009 3E00               LD    A,0              ; 8inches
C00B 325DEF             LD    (0EF5DH),A       ; drive type 8inches DMA
C00E CD673D             CALL  3D67H            ; subroutine data set
                ;
C011 1602               LD    D,2              ; top of track
                ;
C013 CD45C0     BACKL:  CALL  TRKDSP
                ;
C016 7A                 LD    A,D              ; get head
C017 E601               AND   1
C019 47                 LD    B,A
                ;
C01A 0E00               LD    C,0              ; drive 0
C01C 3E03               LD    A,3              ; read
C01E CD54C0             CALL  SETPRM           ; parameter set
                ;
C021 C5                 PUSH  BC
C022 D5                 PUSH  DE
C023 CD7539             CALL  3975H            ; read to buffer
C026 D1                 POP   DE
C027 C1                 POP   BC
C028 3849               JR    C,ERROR
                ;
C02A 0E01               LD    C,1              ; drive 1
C02C 3E02               LD    A,2              ; write
C02E CD54C0             CALL  SETPRM
                ;
C031 D5                 PUSH  DE
C032 CD7539             CALL  3975H            ; write to disk
C035 D1                 POP   DE
C036 383B               JR    C,ERROR
                ;
C038 14                 INC   D
C039 3E98               LD    A,152
C03B BA                 CP    D
C03C 30D5               JR    NC,BACKL
                ;
C03E 21E4C0             LD    HL,COMP
C041 CD7AC0             CALL  DSPMES
C044 C9                 RET
                ;
C045 D5         TRKDSP:PUSH  DE               ; print track number
C046 21D3C0             LD    HL,TRKMES
C049 CD7AC0             CALL  DSPMES
                ;
C04C 6A                 LD    L,D
C04D 2600               LD    H,0
C04F CDC228             CALL  28C2H            ; output integer
                ;
C052 D1                 POP   DE
C053 C9                 RET
                ;
C054 2117EF     SETPRM:LD    HL,0EF17H        ; top of parameter table
C057 77                 LD    (HL),A           ; set mode
C058 23                 INC   HL
C059 78                 LD    A,B              ; set HD,US1,US0
C05A B7                 OR    A
C05B 07                 RLCA
C05C 07                 RLCA
C05D B1                 OR    C
C05E 77                 LD    (HL),A
C05F 23                 INC   HL
C060 7A                 LD    A,D
C061 B7                 OR    A
C062 1F                 RRA
C063 77                 LD    (HL),A           ; set cylinder
C064 23                 INC   HL
C065 3E01               LD    A,1              ; set sector
C067 77                 LD    (HL),A
C068 23                 INC   HL
C069 3E1A               LD    A,26             ; number of sector
C06B 77                 LD    (HL),A
C06C 21FDC0             LD    HL,BUFFER        ; set data address
C06F 221CEF             LD    (0EF1CH),HL
```

```
C072 C9                 RET
                ;
C073 21F1C0     ERROR: LD   HL,ERRMES           ; print error message
C076 CD7AC0             CALL DSPMES
C079 C9                 RET
                ;
C07A 7E         DSPMES:LD   A,(HL)              ; display message subroutine
C07B A7                 AND  A
C07C C8                 RET  Z
C07D CD0D3E             CALL 3E0DH
C080 23                 INC  HL
C081 18F7               JR   DSPMES
                ;
C083 0D0A496E PROMPT:DEFM 0DH,0AH,'Insert source disk '
C087 73657274
C08B 20736F75
C08F 72636520
C093 6469736B
C097 20
C098 746F2064           DEFM 'to drive 1',0DH,0AH
C09C 72697665
C0A0 20310D0A
C0A4 496E7365           DEFM 'Insert new disk to '
C0A8 7274206E
C0AC 65772064
C0B0 69736B20
C0B4 746F20
C0B7 64726976           DEFM 'drive 2',0DH,0AH
C0BB 6520320D
C0BF 0A
C0C0 616E6420           DEFM 'and hit any key !!',0
C0C4 68697420
C0C8 616E7920
C0CC 6B657920
C0D0 212100
                ;
C0D3 0D0A436F TRKMES:DEFM 0DH,0AH,'Copying track ',0
C0D7 7079696E
C0DB 67207472
C0DF 61636B20
C0E3 00
                ;
C0E4 0D0A436F COMP:  DEFM 0DH,0AH,'Completed.',0
C0E8 6D706C65
C0EC 7465642E
C0F0 00
                ;
C0F1 0D0A4572 ERRMES:DEFM 0DH,0AH,'Error !!',7,0
C0F5 726F7220
C0F9 21210700
                ;
C0FD            BUFFER:DEFS 1A00H
                ;
DAFD                    END
```

```
a=&hc000:call a


Insert source disk to drive 1
Insert new disk to drive 2
and hit any key !!
Copying track 2
Copying track 3


Copying track 151
Copying track 152
Completed.
Ok
```

３Ａ２ＤＨ　ＦＤＣのイニシァライズを行う．

アドレス　３Ａ２ＤＨ

機　能　ＦＤＣのイニシァライズを行う．

レジスタ　Ａ，Ｆ

解　説　８インチ，５インチのＤＭＡ（ダイレクト・メモリ・アクセス）転送方式のＦＤＣ（フロッピ・ディスク・コントローラ）をそれぞれについてイニシァライズし，各ドライブのチェックを行うサブルーチンです．インタフェースの選択（８インチ用，５インチ用）の指定は，ＥＦ５６Ｈ番地にインタフェースのポート・アドレスをセットすることで行い，その値は

８インチ：２０Ｈ，５インチ：１０Ｈ

となっています．また，ワークエリアに他にも８インチ用と５インチ用のデータをどちらかをセットする必要がありますが，それは３Ｄ６７Ｈ番地からのサブルーチンを利用することで実現できます．以上については３Ｄ６７Ｈ番地の項を御参照下さい．

このサブルーチンで行われる操作は以下の様になります．

１）ＳＲＴ（ステップ・レート・タイム），ＨＵＴ（ヘッド・アンロード・タイム），ＨＬＴ（ヘッド・ロード・タイム），ＮＤ（ＤＭＡ non－ＤＭＡモード），各データの設定．

２）各デバイス（ドライブ）のチェック，接続されていれば，ＥＦ４ＢＨ，４ＣＨ番地で示されるアドレスで初まるステータス・テーブル上にＳＴ０（ステータス０），ＳＴ１（ステータス１）の２バイトを各デバイスごとに書き込む．

サンプル

```
                ;
                ; --- FDC initialize and check drive connection
                ;
                ;       sample of subroutine from 3A2DH
                ;
                        ORG   0B900H
                ;
B900 2182B9     FDCINI:LD    HL,INCH8          ; message for 8 inches
B903 CD79B9            CALL  DSPMES
                ;
B906 3E00              LD    A,0               ; 8 inches
B908 325DEF            LD    (0EF5DH),A
B90B CD673D            CALL  3D67H             ; set data for 8 inches
                ;
B90E 3A4EEF            LD    A,(0EF4EH)        ; check interface port
B911 4F                LD    C,A
B912 ED78              IN    A,(C)
B914 CB47              BIT   0,A
B916 2808              JR    Z,FDC8
```

```
                         ;
B918 219AB9                    LD    HL,NODRV             ; no drive
B91B CD79B9                    CALL  DSPMES
                         ;
B91E 180C                      JR    FDC8J
                         ;
B920 CD2D3A              FDC8:  CALL  3A2DH               ; initialize interface of 8´
                         ;
B923 CD65B9                    CALL  COUNT                ; count drives
B926 6F                        LD    L,A
B927 2600                      LD    H,0
B929 CDC228                    CALL  28C2H                ; output integer
                         ;
B92C 219DB9              FDC8J: LD    HL,CONMES
B92F CD79B9                    CALL  DSPMES
                         ;
B932 218EB9              FDC5M: LD    HL,INCH5             ; message for 5 inches
B935 CD79B9                    CALL  DSPMES
                         ;
B938 3E01                      LD    A,1                  ; 5 inches
B93A 325DEF                    LD    (0EF5DH),A
B93D CD673D                    CALL  3D67H                ; set data for 5 inches
                         ;
B940 3A4EEF                    LD    A,(0EF4EH)
B943 4F                        LD    C,A
B944 ED78                      IN    A,(C)
B946 CB47                      BIT   0,A
B948 2808                      JR    Z,FDC5
                         ;
B94A 219AB9                    LD    HL,NODRV
B94D CD79B9                    CALL  DSPMES
                         ;
B950 180C                      JR    FDC5J
                         ;
B952 CD2D3A              FDC5:  CALL  3A2DH               ; initialize interface of 5´
                         ;
B955 CD65B9                    CALL  COUNT
B958 6F                        LD    L,A
B959 2600                      LD    H,0
B95B CDC228                    CALL  28C2H
                         ;
B95E 219DB9              FDC5J: LD    HL,CONMES
B961 CD79B9                    CALL  DSPMES
                         ;
B964 C9                        RET
                         ;
B965 2A4BEF              COUNT: LD    HL,(0EF4BH)          ; address of status table
B968 0E00                      LD    C,0                  ; counter clear
                         ;
B96A 7E                  COUNTL:LD    A,(HL)
B96B FE10                      CP    10H                  ; unconnected drive
B96D 2808                      JR    Z,COUNTJ
                         ;
B96F 0C                        INC   C
B970 23                        INC   HL
B971 23                        INC   HL
B972 79                        LD    A,C
B973 FE04                      CP    4
B975 38F3                      JR    C,COUNTL
                         ;
B977 79                  COUNTJ:LD    A,C
B978 C9                        RET
                         ;
B979 7E                  DSPMES:LD    A,(HL)               ; display message subroutine
B97A A7                        AND   A
B97B C8                        RET   Z
B97C CD0D3E                    CALL  3E0DH
B97F 23                        INC   HL
B980 18F7                      JR    DSPMES
                         ;
B982 0D0A3820            INCH8: DEFM  0DH,0AH,´8 INCHES:´,0
B986 494E4348
B98A 45533A00
B98E 0D0A3520            INCH5: DEFM  0DH,0AH,´5 INCHES:´,0
```

```
B992 494E4348
B996 45533A00
                 ;
B99A 4E6F00      NODRV: DEFM 'No',0
                 ;
B99D 20647269 CONMES:DEFM ' drives are connected.',0
B9A1 76657320
B9A5 61726520
B9A9 636F6E6E
B9AD 65637465
B9B1 642E00
                 ;
B9B4                    END
```

```
a=&hb900:call a

8 INCHES:2 drives are connected.
5 INCHES:No drives are connected.
Ok
```

３Ａ８８Ｈ　ＤＭＡドライブのヘッドのリストア．

アドレス	３Ａ８８Ｈ
機　能	ＤＭＡドライブのヘッドのリストアを行う．
レジスタ	Ａ，Ｆ

解　説　Ｃレジスタの第０，第１ビットで指定されるＤＭＡドライブの，同じくＣレジスタの第２ビットで指定されるヘッドを，トラック０までリストアするサブルーチンです．ここでのドライブナンバはＤＭＡタイプの中における番号で，０からの値で与えます．具体的にはトラック０を指定（Ｄレジスタ＝０）した後，３ＡＡＤＨ番地からのサブルーチンに合流しています．

エラーの発生した場合にはＣＹフラグをセットし，またＥＦ３ＤＨ番地に８２Ｈを格納して戻ります．

サンプル

```
                ;
                ; --- restore head 0 of DMA drive
                ;
                ;       sample of subroutine from 3A88H
                ;
                        ORG  0B900H
                ;
B900 2134B9     RSTHD0:LD   HL,DRIVE            ; input drive number
B903 CD2BB9            CALL DSPMES
B906 CDC85F            CALL 5FC8H               ; line input subroutine
B909 D8                RET  C
                ;
B90A 23                INC  HL
B90B CDBC26            CALL 26BCH               ; convert to binary
B90E CDA021            CALL 21A0H               ; convert to integer
B911 7D                LD   A,L
B912 3D                DEC  A
                ;
B913 4F                LD   C,A
                ;
B914 CD883A            CALL 3A88H               ; head 0 restore
                ;
B917 2146B9            LD   HL,MESS1            ; restored message
B91A CD2BB9            CALL DSPMES
                ;
B91D 79                LD   A,C
B91E C631              ADD  A,'1'
B920 CD0D3E            CALL 3E0DH
                ;
B923 2157B9            LD   HL,MESS2
B926 CD2BB9            CALL DSPMES
                ;
B929 18D5              JR   RSTHD0
                ;
B92B 7E         DSPMES:LD   A,(HL)
B92C A7                AND  A
B92D C8                RET  Z
B92E CD0D3E            CALL 3E0DH
B931 23                INC  HL
B932 18F7              JR   DSPMES
```

```
                 ;
B934 0D0A4472 DRIVE: DEFM 0DH,0AH,'Drive number ? ',0
B938 69766520
B93C 6E756D62
B940 6572203F
B944 2000
B946 48656164 MESS1: DEFM 'Head 0 of drive ',0
B94A 2030206F
B94E 66206472
B952 69766520
B956 00
B957 20726573 MESS2: DEFM '  estored !!',0
B95B 746F7265
B95F 64202121
B963 00
                 ;
B964                    END
```

```
a=&hb900:call a

Drive number ? 1
Head 0 of drive 1 restored !!
Drive number ? 2
Head 0 of drive 2 restored !!
Drive number ?
Ok
```

3AADH　DMAドライブのヘッドのシーク.

アドレス	3AADH
機　能	DMAドライブのヘッドのシーク動作を行う.
レジスタ	A, F

解　説　Cレジスタの第0，第1ビットで指定されるDMAドライブの，同じくCレジスタの第2ビットで指定されるヘッドを,Dレジスタで指定されるシリンダ（トラック）までシークします．ここでのドライブナンバはDMAドライブの中における番号で0からの値で与えます．

エラーの発生した場合には,CYフラグをセットし,またEF3DH番地に82Hをセットして戻ります．

サンプル

```
                ;
                ; --- seek head of DMA type drive
                ;
                ;       sample of subroutine from 3AADH
                ;
                        ORG   0B900H
                ;
B900 2174B9     SEEKHD:LD     HL,DRIVE          ; input drive number
B903 CD54B9             CALL  GETNUM
B906 D8                 RET   C                 ; stop key return
                ;
B907 3D                 DEC   A
B908 4F                 LD    C,A
B909 2186B9             LD    HL,HEAD           ; input head number
B90C CD54B9             CALL  GETNUM
B90F B7                 OR    A                 ; clear carry
B910 07                 RLCA
B911 07                 RLCA
B912 B1                 OR    C
B913 4F                 LD    C,A
                ;
B914 2195B9             LD    HL,CYLIND         ; input cylinder number
B917 CD54B9             CALL  GETNUM
B91A 57                 LD    D,A
                ;
B91B CDAD3A             CALL  3AADH             ; seek head
                ;
B91E 21A8B9             LD    HL,MESS1          ; head message
B921 CD4BB9             CALL  DSPMES
B924 79                 LD    A,C
B925 E604               AND   4                 ; clear carry and mask US
B927 0F                 RRCA
B928 0F                 RRCA
B929 CD69B9             CALL  DSPACC
                ;
B92C 21B0B9             LD    HL,MESS2          ; drive message
B92F CD4BB9             CALL  DSPMES
B932 79                 LD    A,C
B933 E603               AND   3                 ; mask HD
B935 3C                 INC   A
B936 CD69B9             CALL  DSPACC
                ;
```

```
B939 21BBB9           LD    HL,MESS3          ; cylinder message
B93C CD4BB9           CALL  DSPMES
B93F 7A               LD    A,D
B940 CD69B9           CALL  DSPACC
              ;
B943 21D0B9           LD    HL,MESS4
B946 CD4BB9           CALL  DSPMES
              ;
B949 18B5             JR    SEEKHD
              ;
B94B 7E       DSPMES:LD     A,(HL)            ; display message
B94C A7               AND   A
B94D C8               RET   Z
B94E CD0D3E           CALL  3E0DH
B951 23               INC   HL
B952 18F7             JR    DSPMES
              ;
B954 C5       GETNUM:PUSH   BC                ; get number
B955 D5               PUSH  DE
B956 CD4BB9           CALL  DSPMES
B959 CDC85F           CALL  5FC8H             ; subroutine line input
B95C F5               PUSH  AF
B95D 23               INC   HL
B95E CDBC26           CALL  26BCH             ; convert to binary
B961 CDA021           CALL  21A0H             ; convert to integer
B964 F1               POP   AF
B965 7D               LD    A,L
B966 D1               POP   DE
B967 C1               POP   BC
B968 C9               RET
              ;
B969 C5       DSPACC:PUSH   BC                ; display accumulator
B96A D5               PUSH  DE
B96B 6F               LD    L,A
B96C 2600             LD    H,0
B96E CDC228           CALL  28C2H             ; display integer
B971 D1               POP   DE
B972 C1               POP   BC
B973 C9               RET
              ;
B974 0D0A4472 DRIVE:  DEFM  0DH,0AH,'Drive number ? ',0
B978 69766520
B97C 6E756D62
B980 6572203F
B984 2000
B986 48656164 HEAD:   DEFM  'Head number ? ',0
B98A 206E756D
B98E 62657220
B992 3F2000
B995 43796C69 CYLIND:DEFM   'Cylinder number ? ',0
B999 6E646572
B99D 206E756D
B9A1 62657220
B9A5 3F2000
              ;
B9A8 0D0A4865 MESS1:  DEFM  0DH,0AH,'Head ',0
B9AC 61642000
B9B0 206F6620 MESS2:  DEFM  ' of drive ',0
B9B4 64726976
B9B8 652000
B9BB 20736565 MESS3:  DEFM  ' seeked to cylinder ',0
B9BF 6B656420
B9C3 746F2063
B9C7 796C696E
B9CB 64657220
B9CF 00
B9D0 2021210D MESS4:  DEFM  ' !!',0DH,0AH,0
B9D4 0A00
              ;
B9D6                  END
```

```
a=&hb900:call a

Drive number ? 1
Head number ? 0
Cylinder number ? 0

Head 0 of drive 1 seeked to cylinder 0 !!

Drive number ? 2
Head number ? 1
Cylinder number ? 76

Head 1 of drive 2 seeked to cylinder 76 !!

Drive number ?
Ok
```

３ＡＦ７Ｈ　ＦＤＣステータスの読み込みを行う．

アドレス　３ＡＦ７Ｈ

機　　能　ＦＤＣステータスを読み込みメモリに格納する．

レジスタ　Ａ，Ｆ

解　　説　ＦＤＣ（フロッピ・ディスク・コントローラ）のレザルト・フェーズ（ＲＰ）において読み込まれるＦＤＣのステータスを，ＥＦ２６Ｈ番地からのワークエリアに格納するサブルーチンです．パラメータとしては，ＦＤＣに動作を要求したときのドライブ・ナンバ，ヘッド・ナンバをＣレジスタに与えますが，そのビット割り当ては次の様になっています．

第０・第１ビット………ドライブ・ナンバ（０～）
第２ビット　　　………ヘッド番号（０，１）

また，このサブルーチンから戻ったときにＣＹフラグがセットされていれば，ＦＤＣの動作中にエラーの発生したことを示します．

受け取るステータス・バイトは全部で７バイトあり，各バイトの名称と意味は以下の様になります．

ＥＦ２６Ｈ　ＳＴ０………ステータス０
ＥＦ２７Ｈ　ＳＴ１………ステータス１
ＥＦ２８Ｈ　ＳＴ２………ステータス２
ＥＦ２９Ｈ　　Ｃ　………コマンド実行終了時のシリンダ・ナンバ
ＥＦ２ＡＨ　　Ｈ　………コマンド実行終了時のヘッド・ナンバ
ＥＦ２ＢＨ　　Ｒ　………コマンド実行終了時のセクタ・ナンバ
ＥＦ２ＣＨ　　Ｎ　………コマンド実行終了時の１セクタあたりのデータ長

なお，各データの詳しい内容については，ＦＤＣの説明の項で詳述します．

３Ｂ２６Ｈ　データマージンを改定する．

アドレス	３Ｂ２６Ｈ
機　　能	データマージンを改定する．
レジスタ	Ａ，Ｆ

解　　説　ＤＭＡ（ダイレクト・メモリ・アクセス）転送方式のディスクドライブに対してリード／ライト系のコマンドを実行しエラーの発生した場合には，データマージンを変化させて最適なデータマージンの値の見つかるまでリトライを繰り返す必要があります．このサブルーチンでは，現在設定されているデータマージンの値を指すポインタを＋１し，次の新しいデータマージンの値を設定します．そして，全ての異なるデータマージンを設定しきると，再び最初のデータマージンの値から順に設定してゆきます．

　データマージンの値は，３Ｂ５８Ｈ～３Ｂ６１Ｈ番地に順に格納されており，このサブルーチンではこの領域のデータを順に取り出しています．また，設定するデータは各ドライブごとに適した値を出力する必要があるため，各ドライブへの出力データをポイントしたアドレスを，ドライブごとにワークエリア（ＥＦ４１Ｈ～ＥＦ４８Ｈ）に保持しておく必要があります．これらのアドレスはこのサブルーチンをＣＡＬＬする際に自動的に改定され，それらはドライブごとに以下の様に割り当てられています．

ドライブ０：ＥＦ４１Ｈ，４２Ｈ
ドライブ１：ＥＦ４３Ｈ，４４Ｈ
ドライブ２：ＥＦ４５Ｈ，４６Ｈ
ドライブ３：ＥＦ４７Ｈ，４８Ｈ

　ここでいうドライブ・ナンバとは，実際のドライブ・ナンバより１を引いたもので，Ｃレジスタにより指定します．また，マージンデータ出力用のポートアドレスをＥＦ４ＤＨ番地にあらかじめセットしておく必要があり，使用するＤＭＡ転送方式のドライブが８インチの場合はＦ５Ｈ，５インチの場合はＦ９Ｈとなっています．

３Ｂ６２Ｈ　データマージンを設定する．

アドレス　３Ｂ６２Ｈ

機　能　データマージンを設定する．

レジスタ　Ａ，Ｆ

解　説　Ｃレジスタの値（０～）で示されるＤＭＡ（ダイレクト・メモリアクセス）転送方式のディスクドライブに対する，データマージンの値を設定するサブルーチンです．設定するデータは，各ドライブによって適した値を出力する必要があるため，各ドライブへの出力データをポイントしたアドレスを，ドライブごとにワークエリア（ＥＦ４１Ｈ～ＥＦ４８Ｈ）に保持しています．各ドライブのポインタは，それぞれ以下のアドレスに格納されています．

ドライブ０：　ＥＦ４１Ｈ，４２Ｈ
ドライブ１：　ＥＦ４３Ｈ，４４Ｈ
ドライブ２：　ＥＦ４５Ｈ，４６Ｈ
ドライブ３：　ＥＦ４７Ｈ，４８Ｈ

ここでいうドライブ・ナンバには，実際のドライブ・ナンバより１を引いたものを使用します．また，マージンデータ出力用のポートアドレスをＥＦ４ＤＨにあらかじめセットしておく必要があり，使用するＤＭＡ転送方式のドライブが８インチの場合はＦ５Ｈ，５インチの場合はＦ９Ｈとなっています．

3BAEH　FDCにパラメータを送信する.

アドレス	3BAEH
機　能	FDCにパラメータを送信する.
レジスタ	A，F

解　説　FDC（フロッピ・ディスク・コントローラ）のコマンド・フェーズ(CP)において，必要なパラメータをまとめてFDCに送信するサブルーチンです．コール時に必要なパラメータは以下の様になります．

Cレジスタ………HD，US1，US0合成バイト（第2ビット：ヘッド・ナンバ，第1，第0ビット：ドライブ・ナンバ（0〜3）．

Dレジスタ………シリンダ・ナンバ．

Eレジスタ………セクタ・ナンバ．

EF52H………GPL（ギャップ長）値．

また，実際に送信されるパラメータは以下の様になります．

HD，US1，US0（ヘッド・ナンバ，ドライブ・ナンバ）
C（シリンダ・ナンバ）
H（ヘッド・ナンバ）
R（セクタ・ナンバ）
N（1セクタあたりのデータ長）
EOT（1トラックの最終セクタ・ナンバ）
GPL（ギャップ長）
DTL（データ長，通常FFH）

なお，このサブルーチンは，FDCのコマンド・フェーズにおけるパラメータ送信のフォーマットが上記の様になっているものしか使用できません．

３Ｃ７ＦＨ　ＦＤＣからデータを受信する.

アドレス	３Ｃ７ＦＨ
機　能	ＦＤＣからデータを受信する.
レジスタ	Ａ，Ｆ

解　説　ＦＤＣ（フロッピ・ディスク・コントローラ）から１バイトのデータを受け取り，アキュムレータに格納して戻るサブルーチンです．このサブルーチンでは，ＦＤＣのステータス・レジスタをチェックし，ＢＩＯビットが１，ＲＱＭビットが１になるまでループを行い，これらの条件が成立するとＦＤＣよりデータを入力します．

インタフェースの選択（８インチ用，５インチ用）については，ワークエリアにセットする入出力ポートのアドレス・ナンバを換えることで行い，以下の様にセットします．

ＥＦ５４Ｈ番地………ステータス・レジスタ入力用ポートのアドレス.
（８インチ：Ｆ６Ｈ，５インチ：ＦＡＨ）
ＥＦ５５Ｈ番地………データ・レジスタ入力用ポートのアドレス.
（８インチ：Ｆ７Ｈ，５インチ：ＦＢＨ）

サンプル

```
                ;
                ; --- check device status
                ;
                ;       sample of subroutine from 3C7FH
                ;
                        ORG   0B900H
                ;
B900 2184B9     CHKDEV:LD    HL,DRIVE           ; input drive number
B903 CD7BB9            CALL  DSPMES
B906 CDC85F            CALL  5FC8H              ; line input from keyboard
B909 D8                RET   C                  ; stop key return
                ;
B90A 23                INC   HL
B90B CDBC26            CALL  26BCH              ; convert to binary
B90E CDA021            CALL  21A0H              ; convert to integer
B911 7D                LD    A,L
                ;
B912 3D                DEC   A
B913 CD5899            CALL  9958H              ; set disk identification
                ;
B916 3A5DEF            LD    A,(0EF5DH)         ; check drive type
B919 01F6F7            LD    BC,0F7F6H          ; port for 8 inches
B91C B7                OR    A
B91D 2807              JR    Z,CHKDVM
                ;
B91F 01FAFB            LD    BC,0FBFAH          ; port for 5 inches
B922 FE01              CP    1
B924 20DA              JR    NZ,CHKDEV
                ;
B926 ED4354EF   CHKDVM:LD    (0EF54H),BC
B92A 3E04              LD    A,4                ; SENSE DEVICE STATUS command
B92C CD943C            CALL  3C94H              ; send command to FDC
```

```
                        ;
B92F CDD93D                     CALL 3DD9H              ; get logical drive number
B932 325FEF                     LD   (0EF5FH),A
                        ;
B935 CD943C                     CALL 3C94H              ; send device number
                        ;
B938 CD7F3C                     CALL 3C7FH              ; recieve data from FDC
                        ;
B93B CB6F                       BIT  5,A                ; check ready
B93D CC5AB9                     CALL Z,NOTRDY
B940 280D                       JR   Z,CHKDVE
                        ;
B942 CB77                       BIT  6,A                ; check write protect
B944 C451B9                     CALL NZ,WRTPRO
                        ;
B947 CB5F                       BIT  3,A                ; check two sides
B949 C463B9                     CALL NZ,DBLSD
B94C CC6CB9                     CALL Z,SNGSD
                        ;
B94F 18AF               CHKDVE:JR   CHKDEV
                        ;
B951 F5                 WRTPRO:PUSH AF
B952 2196B9                     LD   HL,WPMES           ; write protect disk
B955 CD7BB9                     CALL DSPMES
B958 F1                         POP  AF
B959 C9                         RET
                        ;
B95A F5                 NOTRDY:PUSH AF                  ; drive not ready
B95B 21ABB9                     LD   HL,NDMES
B95E CD7BB9                     CALL DSPMES
B961 F1                         POP  AF
B962 C9                         RET
                        ;
B963 F5                 DBLSD: PUSH AF                  ; double sided media
B964 21C1B9                     LD   HL,DSMES
B967 CD7BB9                     CALL DSPMES
B96A 1807                       JR   SNGSDJ
                        ;
B96C F5                 SNGSD: PUSH AF                  ; single sided media
B96D 21C8B9                     LD   HL,SSMES
B970 CD7BB9                     CALL DSPMES
                        ;
B973 21CFB9             SNGSDJ:LD   HL,MDMES
B976 CD7BB9                     CALL DSPMES
B979 F1                         POP  AF
B97A C9                         RET
                        ;
B97B 7E                 DSPMES:LD   A,(HL)              ; display message subroutine
B97C A7                         AND  A
B97D C8                         RET  Z
B97E CD0D3E                     CALL 3E0DH
B981 23                         INC  HL
B982 18F7                       JR   DSPMES
                        ;
B984 0D0A4472 DRIVE: DEFM 0DH,0AH,'Drive number ? ',0
B988 69766520
B98C 6E756D62
B990 6572203F
B994 2000
                        ;
B996 57726974 WPMES: DEFM 'Write protected !!',0DH,0AH,0
B99A 65207072
B99E 6F746563
B9A2 74656420
B9A6 21210D0A
B9AA 00
B9AB 44657669 NDMES: DEFM 'Device not ready !!',0DH,0AH,0
B9AF 6365206E
B9B3 6F742072
B9B7 65616479
B9BB 2021210D
B9BF 0A00
                        ;
B9C1 446F7562 DSMES: DEFM 'Double',0
```

```
B9C5 6C6500
B9C8 53696E67 SSMES: DEFM 'Single',0
B9CC 6C6500
B9CF 20736964 MDMES: DEFM ' sided media.',0DH,0AH,0
B9D3 6564206D
B9D7 65646961
B9DB 2E0D0A00
                ;
B9DF                    END
```

```
a=&hb900:call a


Drive number ? 1
Double sided media.

Drive number ? 2
Device not ready !!

Drive number ? 3
Write protected !!
Double sided media.

Drive number ?
Ok
```

３Ｃ９４Ｈ　ＦＤＣにデータを送信する．

アドレス　３Ｃ９４Ｈ

機　　能　ＦＤＣにデータを送信する．

レジスタ　全て待避

解　　説　ＦＤＣ（フロッピ・ディスク・コントローラ）へ，アキュムレータのデータを送信し戻るサブルーチンです．このサブルーチンでは，ＦＤＣのステータス・レジスタをチェックし，ＤＩＯビットが０，ＲＱＭビットが１になるまでループを行い，それらの条件が成立するとＦＤＣへデータを出力します．

インタフェースの選択（８インチ用，５インチ用）については，ワークエリアにセットする入出力ポートのアドレス・ナンバを換えることで行い，以下の様にセットします．

ＥＦ５４Ｈ番地………ステータス・レジスタ入力用ポートのアドレス．
（８インチ：Ｆ６Ｈ，５インチ：ＦＡＨ）

ＥＦ５５Ｈ番地………データ・レジスタ出力用ポートのアドレス．
（８インチ：Ｆ７Ｈ，５インチ：ＦＢＨ）

サンプル

```
                ;
                ; --- 8 inch DMA type disk physical format
                ;
                ;       sample of subroutine from 3C94H
                ;
                        ORG     0C000H
                ;
C000 21FBC0     FORMAT:LD       HL,FMTMES       ; message of in format
C003 CDF2C0             CALL    DSPMES
                ;
C006 3E01               LD      A,1             ; set top of cylinder
C008 3261C1             LD      (CYLIND),A
                ;
C00B 3E00       CYLINL:LD       A,0             ; set head
C00D 3262C1             LD      (HEADNM),A
                ;
C010 3E08               LD      A,8             ; mergin control
C012 D3F5               OUT     (0F5H),A
                ;
                ; create sque table
                ;
C014 2163C1     HEADL: LD       HL,STABLE       ; top of sque table area
C017 1147C1             LD      DE,SQUTBL       ; top of sector sque table
C01A 061A               LD      B,26            ; sector a cylinder
                ;
C01C 3A61C1     CRTBLL:LD       A,(CYLIND)      ; set C
C01F 77                 LD      (HL),A
C020 23                 INC     HL
C021 3A62C1             LD      A,(HEADNM)      ; set H
C024 77                 LD      (HL),A
C025 23                 INC     HL
C026 1A                 LD      A,(DE)          ; set R
C027 77                 LD      (HL),A
C028 23                 INC     HL
```

```
C029 3E01          LD   A,1             ; set N
C02B 77            LD   (HL),A
C02C 23            INC  HL
C02D 13            INC  DE
C02E 10EC          DJNZ CRTBLL
              ;
              ; seek head
              ;
C030 3E07          LD   A,7             ; set flag for interrupt
C032 323EEF        LD   (0EF3EH),A
C035 3E0F          LD   A,0FH           ; SEEK command
C037 CD943C        CALL 3C94H           ; subroutine send command
C03A 3A62C1        LD   A,(HEADNM)      ; send HD,US1,US0
C03D B7            OR   A
C03E 07            RLCA
C03F 07            RLCA
C040 F601          OR   1
C042 CD943C        CALL 3C94H
C045 3A61C1        LD   A,(CYLIND)
C048 CD943C        CALL 3C94H
              ;
C04B CD713C        CALL 3C71H           ; wait interrupt
              ;
C04E 3A26EF        LD   A,(0EF26H)      ; ST0
C051 E6E0          AND  0E0H            ; check seek end
C053 FE20          CP   20H             ; check normal
C055 C2DEC0        JP   NZ,ERRORS
              ;
              ; control DMAC
              ;
C058 2163C1        LD   HL,STABLE
C05B 7D            LD   A,L             ; send lower address
C05C D362          OUT  (62H),A
C05E 7C            LD   A,H             ; send upper address
C05F D362          OUT  (62H),A
C061 3E67          LD   A,103           ; send lower bytes
C063 D363          OUT  (63H),A
C065 3E80          LD   A,80H+0         ; send upper bytes
C067 D363          OUT  (63H),A
C069 3EA6          LD   A,0A6H          ; send mode and DMA start
C06B D368          OUT  (68H),A
              ;
              ; select interface
              ;
C06D 3E02          LD   A,2             ; select interface for 8 inches
C06F D3F3          OUT  (0F3H),A
              ;
              ; motor control
              ;
C071 3A61C1        LD   A,(CYLIND)
C074 FE26          CP   38              ; max cylinder
C076 3E0F          LD   A,0FH
C078 3802          JR   C,MTCNTJ
C07A 3E3F          LD   A,3FH
C07C D3F4     MTCNTJ:OUT (0F4H),A
              ;
              ; write ID
              ;
C07E 3EFF          LD   A,0FFH          ; set flag for interrupt
C080 323EEF        LD   (0EF3EH),A
              ;
C083 3E4D          LD   A,4DH           ; write ID command MFM
C085 CD943C        CALL 3C94H           ; subroutine send command
C088 3A62C1        LD   A,(HEADNM)      ; send HD,US1,US0
C08B B7            OR   A
C08C 07            RLCA
C08D 07            RLCA
C08E F601          OR   1               ; appoint drive 2
C090 CD943C        CALL 3C94H
C093 3E01          LD   A,1             ; send N
C095 CD943C        CALL 3C94H
C098 3E1A          LD   A,26            ; send SC
C09A CD943C        CALL 3C94H
C09D 3E36          LD   A,36H           ; send GPL
```

```
C09F CD943C          CALL 3C94H
C0A2 3EE5            LD   A,0E5H          ; send D
C0A4 CD943C          CALL 3C94H
               ;
C0A7 CD713C          CALL 3C71H           ; wait interrupt from FDC
               ;
               ; check result status
               ;
C0AA 3EA5            LD   A,0A5H          ; stop DMA
C0AC D368            OUT  (68H),A
C0AE AF              XOR  A               ; deselect interface
C0AF D3F3            OUT  (0F3H),A
               ;
C0B1 3A62C1          LD   A,(HEADNM)
C0B4 B7              OR   A
C0B5 07              RLCA
C0B6 07              RLCA
C0B7 F601            OR   1
C0B9 4F              LD   C,A
C0BA CDF73A          CALL 3AF7H           ; recieve result status
               ;
C0BD 3827            JR   C,ERRORF
               ;
C0BF 3A62C1          LD   A,(HEADNM)      ; head increment
C0C2 3C              INC  A
C0C3 3262C1          LD   (HEADNM),A
C0C6 FE01            CP   1
C0C8 CA14C0          JP   Z,HEADL
               ;
C0CB 3A61C1          LD   A,(CYLIND)      ; cylinder increment
C0CE 3C              INC  A
C0CF 3261C1          LD   (CYLIND),A
C0D2 FE27            CP   39
C0D4 DA0BC0          JP   C,CYLINL
               ;
C0D7 210FC1          LD   HL,COMP         ; message for complete
C0DA CDF2C0          CALL DSPMES
               ;
C0DD C9              RET
               ;
C0DE 2123C1    ERRORS:LD  HL,ERSMES       ; output error message of seek
C0E1 CDF2C0          CALL DSPMES
C0E4 1806            JR   ERRORJ
               ;
C0E6 2134C1    ERRORF:LD  HL,ERFMES       ; output error of format
C0E9 CDF2C0          CALL DSPMES
               ;
C0EC 3E80      ERRORJ:LD  A,80H           ; error flag set
C0EE 323DEF          LD   (0EF3DH),A
C0F1 C9              RET
               ;
C0F2 7E        DSPMES:LD  A,(HL)          ; display message subroutine
C0F3 A7              AND  A
C0F4 C8              RET  Z
C0F5 CD0D3E          CALL 3E0DH
C0F8 23              INC  HL
C0F9 18F7            JR   DSPMES
               ;
C0FB 0D0A4E6F  FMTMES:DEFM 0DH,0AH,'Now formatting...',0
C0FF 7720666F
C103 726D6174
C107 74696E67
C10B 2E2E2E00
               ;
C10F 0D0A466F  COMP:  DEFM 0DH,0AH,'Format completed.',0
C113 726D6174
C117 20636F6D
C11B 706C6574
C11F 65642E00
               ;
C123 0D0A5365  ERSMES:DEFM 0DH,0AH,'Seek error !!',7,0
C127 656B2065
C12B 72726F72
C12F 20212107
```

```
C133 00
                ;
C134 0D0A466F ERFMES:DEFM 0DH,0AH,'Format error !!',7,0
C138 726D6174
C13C 20657272
C140 6F722021
C144 210700
                ;
                ; sque table for NEC disk
                ;
C147 010E020F SQUTBL:DEFB 1,14,2,15,3,16,4,17,5,18,6,19,7,20,8,21
C14B 03100411
C14F 05120613
C153 07140815
C157 09160A17        DEFB 9.22,10,23,11,24,12,25,13,26
C15B 0B180C19
C15F 0D1A
                ;
C161          CYLIND:DEFS 1
                ;
C162          HEADNM:DEFS 1
                ;
C163          STABLE:DEFS 104
                ;
C1CB                 END
```

```
a=&hc000:call a


Now formatting...
Format completed.
Ok
```

３Ｄ６７Ｈ　ＤＭＡ・ＦＤＤ用データをワークエリアにセットする．

アドレス　３Ｄ６７Ｈ

機　　能　ＤＭＡ・ＦＤＤ用データをワークエリアにセットする．

レジスタ　Ａ，Ｆ，Ｂ，Ｃ，Ｄ，Ｅ，Ｈ，Ｌ

解　　説　ＤＭＡ転送方式のディスクドライブ（８インチ，５インチ）のどちらかを選択するためのデータをワークエリアにセットするためのサブルーチンで，ドライブ・タイプの指定はＥＦ５ＤＨ番地に００Ｈか０１Ｈの値をセットすることで行います．

８インチ：００Ｈ，５インチ：０１Ｈ

また，セットされるデータは，ＥＦ４ＢＨよりＥＦ５ＣＨに格納されます．

ＥＦ４ＢＨ，４ＣＨ
………ドライブ・ステータステーブルのアドレス．

ＥＦ４ＤＨ………マージンコントロール用ポートのアドレス．

ＥＦ４ＥＨ………インタフェース接続チェック用ポートのアドレス．

ＥＦ４ＦＨ………ＦＤＣ（フロッピ・ディスク・コントローラ）制御用ポートのアドレス．

ＥＦ５０Ｈ………最大シリンダ・ナンバ＋１．

ＥＦ５１Ｈ………センタ・シリンダ・ナンバ．

ＥＦ５２Ｈ………最大セクタ・ナンバ．

ＥＦ５３Ｈ………ＧＰＬ（ギャップの長さ）値．

ＥＦ５４Ｈ……… ＦＤＣ・ステータス・レジスタ入力用ポートのアドレス．

ＥＦ５５Ｈ……… ＦＤＣ・データ・レジスタ入出力用ポートのアドレス．

ＥＦ５６Ｈ………ＦＤＣ・インタフェース・セレクト用データ．

ＥＦ５７Ｈ………ＳＲＴ（ステップ・レート・タイム），ＨＵＴ（ヘッド・アンロード・タイム）

ＥＦ５８Ｈ………ＨＬＴ（ヘッド・ロード・タイム），ＮＤ（ＤＭＡスイッチ）

ＥＦ５９Ｈ………ＤＭＡＣ（ＤＭＡコントローラ）制御用コマンド．

ＥＦ５ＡＨ………ＤＭＡＣ・チャンネル選択用ポートのアドレス．

ＥＦ５ＢＨ………ＤＭＡＣ・ベース・アドレス．

ＥＦ５ＣＨ………ＦＤＣ・インタフェース・イネーブルデータ．

ＥＦ５ＣＨ………ＦＤＣ・インタフェース・イネーブルデータ.

サンプル

```
                  ;
                  ; --- check connection of interface
                  ;
                  ;       sample of subroutine from 3D67H
                  ;
                          ORG   0B900H
                  ;
B900 3E00         CHKINT:LD    A,0                  ; 8 inches
B902 325DEF              LD    (0EF5DH),A
B905 CD673D              CALL  3D67H                ; set data for 8 inches
                  ;
B908 3A4EEF              LD    A,(0EF4EH)           ; interface check port
B90B 4F                  LD    C,A
B90C ED78                IN    A,(C)
B90E CB47                BIT   0,A
B910 CC27B9              CALL  Z,CONNE8             ; connection of 8 inches
                  ;
B913 3E01                LD    A,1                  ; 5 inches
B915 325DEF              LD    (0EF5DH),A
B918 CD673D              CALL  3D67H                ; set data for 5 inches
                  ;
B91B 3A4EEF              LD    A,(0EF4EH)
B91E 4F                  LD    C,A
B91F ED78                IN    A,(C)
B921 CB47                BIT   0,A
B923 CC2EB9              CALL  Z,CONNE5             ; connection of 5 inches
                  ;
B926 C9                  RET
                  ;
B927 3E38         CONNE8:LD    A,'8'                ; message for 8 inches
B929 CD0D3E              CALL  3E0DH
B92C 1805                JR    CONNE
                  ;
B92E 3E35         CONNE5:LD    A,'5'                ; message for 5 inches
B930 CD0D3E              CALL  3E0DH
                  ;
B933 2143B9       CONNE: LD    HL,MESS
B936 CD3AB9              CALL  DSPMES
B939 C9                  RET
                  ;
B93A 7E           DSPMES:LD    A,(HL)               ; display message subroutine
B93B A7                  AND   A
B93C C8                  RET   Z
B93D CD0D3E              CALL  3E0DH
B940 23                  INC   HL
B941 18F7                JR    DSPMES
                  ;
B943 20696E63     MESS:  DEFM  ' inches drives '
B947 68657320
B94B 64726976
B94F 657320
B952 61726520            DEFM  'are connected.',0DH,0AH,0
B956 636F6E6E
B95A 65637465
B95E 642E0D0A
B962 00
                  ;
B963                     END
```

```
a=&hb900:call a
8 inches drives are connected.
Ok
```

3DCBH　ドライブタイプを得る.

アドレス	3DCBH
機　能	指定ドライブのタイプを得る.
レジスタ	A, F, H, L

解　説　アキュムレータの値＋1で示されるドライブのタイプを求め，結果をアキュムレータ及びEF5DH番地に格納して戻ります．結果の値とドライブのタイプの関係は次のようになります．

0………DMA 転送方式 8 インチ
1………DMA 転送方式 5 インチ
2………インテリジェントタイプ・片面
3………インテリジェントタイプ・両面

具体的には，EF64H番地からのテーブルを参照してドライブのタイプを得ています．

サンプル

```
                ;
                ; --- get drive type
                ;
                ;        sample of subroutine from 3DCBH
                ;
                        ORG   0B900H
                ;
B900 2138B9     DRVTYP:LD    HL,DRIVE           ; input drive number
B903 CD27B9            CALL  DSPMES
B906 CDC85F            CALL  5FC8H              ; line input subroutine
B909 D8                RET   C                  ; stop key return
                ;
B90A 23                INC   HL
B90B CDBC26            CALL  26BCH              ; convert to binary
B90E CDA021            CALL  21A0H              ; convert to integer
B911 7D                LD    A,L
B912 3D                DEC   A
                ;
B913 CDCB3D            CALL  3DCBH              ; get drive type
                ;
B916 5F                LD    E,A
B917 1600              LD    D,0
B919 2130B9            LD    HL,MESTBL          ; table of address of message
B91C 19                ADD   HL,DE
B91D 19                ADD   HL,DE
B91E 5E                LD    E,(HL)
B91F 23                INC   HL
B920 56                LD    D,(HL)
B921 EB                EX    DE,HL
                ;
B922 CD27B9            CALL  DSPMES
                ;
B925 18D9              JR    DRVTYP
                ;
B927 7E         DSPMES:LD    A,(HL)             ; display message
B928 A7                AND   A
```

```
B929 C8                 RET  Z
B92A CD0D3E             CALL 3E0DH
B92D 23                 INC  HL
B92E 18F7               JR   DSPMES
               ;
B930 4AB9      MESTBL:DEFW DMASTD              ; address of DMA standard
B932 5BB9             DEFW DMAMIN              ; address of DMA mini
B934 6CB9             DEFW INTELS              ; address of inteligent S
B936 8AB9             DEFW INTELD              ; address of inteligent D
               ;
B938 0D0A4472 DRIVE: DEFM 0DH,0AH,'Drive number ? ',0
B93C 69766520
B940 6E756D62
B944 6572203F
B948 2000
               ;
B94A 444D4120 DMASTD:DEFM 'DMA 8 inches !',0DH,0AH,0
B94E 3820696E
B952 63686573
B956 20210D0A
B95A 00
B95B 444D4120 DMAMIN:DEFM 'DMA 5 inches !',0DH,0AH,0
B95F 3520696E
B963 63686573
B967 20210D0A
B96B 00
B96C 496E7465 INTELS:DEFM 'Inteligent single surface !',0DH,0AH,0
B970 6C696765
B974 6E742073
B978 696E676C
B97C 65207375
B980 72666163
B984 6520210D
B988 0A00
B98A 496E7465 INTELD:DEFM 'Inteligent double surface !',0DH,0AH,0
B98E 6C696765
B992 6E742064
B996 6F75626C
B99A 65207375
B99E 72666163
B9A2 6520210D
B9A6 0A00
               ;
B9A8                    END
```

```
a=&hb900:call a

Drive number ? 1
Inteligent double surface !

Drive number ? 2
Inteligent double surface !

Drive number ?
Ok
```

3DD9H　物理ドライブ・ナンバを求める．

アドレス	3DD9H
機　　能	同一ドライブ・タイプ内におけるドライブ・ナンバを求める．
レジスタ	A，F，H，L

解　　説　EC85H番地にドライブ・ナンバ（0～）を与えてコールすることにより，アキュムレータに指定ドライブの同一タイプ内におけるドライブ・ナンバを得ることができます．

具体的には，EC85H番地の値から，上位ドライブの数を引いたものをアキュムレータに与えます．各ドライブ・タイプの順位は以下の様になっています．

1．DMA転送方式8インチ．
2．DMA転送方式5インチ．
3．インテリジェントタイプ・ディスクユニット．

サンプル

```
               ;
               ; --- get number of PC-8031
               ;
               ;       sample of subroutine from 3DD9H
               ;
                       ORG   0B900H
               ;
B900 3A7DEC    GETINT:LD     A,(0EC7DH)         ; maximum drive number
B903 3285EC           LD     (0EC85H),A         ; set number of all drives
               ;
B906 CDD93D           CALL   3DD9H              ; get number of PC-8031
               ;
B909 F5               PUSH   AF                 ; line feed
B90A 3E0D             LD     A,0DH
B90C CD0D3E           CALL   3E0DH
B90F 3E0A             LD     A,0AH
B911 CD0D3E           CALL   3E0DH
B914 F1               POP    AF
               ;
B915 6F               LD     L,A
B916 2600             LD     H,0
B918 CDC228           CALL   28C2H              ; display integer
               ;
B91B 212BB9           LD     HL,MESS
B91E CD22B9           CALL   DSPMES
               ;
B921 C9               RET
               ;
B922 7E        DSPMES:LD     A,(HL)             ; subroutine display message
B923 A7               AND    A
B924 C8               RET    Z
B925 CD0D3E           CALL   3E0DH
B928 23               INC    HL
B929 18F7             JR     DSPMES
               ;
B92B 20647269  MESS:  DEFM   ' drives are inteligent '
B92F 76657320
```

```
B933 61726520
B937 696E7465
B93B 6C696765
B93F 6E7420
B942 74797065        DEFM 'type !!',0DH,0AH,0
B946 2021210D
B94A 0A00
              ;
B94C                 END
```

```
a=&hb900:call a

2 drives are inteligent type !!
Ok
```

〔μＰＤ７６５について〕

ＰＣ－８８０１には，８インチ・５インチの両方のタイプのＤＭＡ（ダイレクト・メモリ・アクセス）転送方式のディスクドライブ（ＰＣ－８８８１など）を接続することができますが，それらのコントロールには高性能なフロッピ・ディスク・コントローラ（ＦＤＣ）であるμＰＤ７６５が使用されています．また，N88－ＢＡＳＩＣにはμＰＤ７６５をコントロールするためのルーチンが用意されており，それらのルーチンを利用することでＤＭＡ転送方式のディスクドライブのコントロールはほとんどの事が可能です．ここでは，μＰＤ７６５について簡単に説明することにします．

３－Ａ－１　μＰＤ７６５の基本動作

このＬＳＩは，他の多くのＬＳＩがそうであるように内部制御プログラムにより動作し，ＣＰＵはＬＳＩに対しコマンドを送ることで間接的にディスクドライブをコントロールします．以下にμＰＤ７６５の主な特徴を列挙します．

１）ＩＢＭディスケット（１，２，２Ｄ）コンパチブル

μＰＤ７６５はＩＢＭ発表のデータ・フォーマットのディスケット（１，２，２Ｄ），すなわち，片面単密度から両面倍密度までのデータ処理を行うこどが可能ですまた，１セクタあたりのデータ長も任意に設定することが可能です

２）マルチセクタ・マルチトラック機能

１トラック内の複数セクタ，または両面ディスクにおける同一シリンダの表裏を１度に処理することができます．

３）ディスクドライブが４台まで接続可能

４）１２８バイト／セクタのディスクにおける部分処理が可能

１セクタ内で処理するデータの数を１２８バイト／セクタのディスクの場合のみで指定できます．

５）ステップ速度・ヘッドロード，アンロード時間プログラマブル

定数の設定がプログラマブルなため，あらゆるディスクドライブに対応．

６）データスキャン機能

条件に合致したセクタを検出することが可能です．

７）データ転送方式プログラマブル

ＤＭＡ転送方式・割り込み方式（non－ＤＭＡ）の一方を選択します．

3－A－2　μＰＤ７６５とのインタフェース

μＰＤ７６５には，メインシステムとのデータ送受用として２種類のレジスタが用意されています．それらはそれぞれデータレジスタ，スラータスレジスタと呼ばれており，それぞれ以下の様な役目を持っています．

１）データレジスタ

μＰＤ７６５とメインシステム間において転送する各種情報（コマンド・パラメータ・レザルトステータスなど）を一時的にストアしておくためのレジスタです．このレジスタに対する書き込み・読み出しの区別は次のステータスレジスタのＤＩＯ，ＲＱＭビットを調べることで行います．

２）ステータスレジスタ

現在ＦＤＣがどの様な状態にあるかを知るためのレジスタで，メインシステムは任意の時点で，その内容を知ることができます．このレジスタのビット構成は Fig.３－Ａ－１の様になっています．

Fig. 3－A－1　ステータスレジスタ

ビット番号	名　称	略　称	内　　容
D 0	FD0 Busy	D 0 B	デバイス＃0がＳＥＥＫコマンドによるシーク動作を実行中であるか，シーク動作終了の割込み要求を保留中であることを示します.
D 1	FD1 Busy	D 1 B	デバイス＃1について0ビットの内容と同様.
D 2	FD2 Busy	D 2 B	デバイス＃2についてＤ0ビットの内容と同様.
D 3	FD3 Busy	D 3 B	デバイス＃3についてＤ0ビットの内容と同様.
D 4	FDC Busy	C B	ＦＤＣがCommand Phase, Result Phase, またはリード／ライト・コマンドのExecution Phaseを実行中であることを示します．このビットがセットされているときは他のコマンドは受付けられません.
D 5	Non DMA MODE	N D M	ＦＤＣがＮｏｎ ＤＭＡモードでデータ転送中であり，メインシステムに対してサービスを要求していることを示します.
D 6	Data Input/Output	D I O	データレジスタを介して転送するデータの方向を示します．0のときはメインシステムからＦＤＣの方向，1のときはＦＤＣからメインシステムの方向を示します．なお，データレジスタの状態はＲＱＭ（Ｄ7ビット）が示します.
D 7	Request for Master	R Q M	ＦＤＣからメインシステムへ転送すべきデータがデータレジスタにロードされていること，またはデータレジスタが空で，メインシステムからＦＤＣへ転送するデータをデータレジスタに書込んでもよいことを示します．データの方向を示すＤＩＯ（Ｄ6ビット）の状態により，次の働きをします. ＤＩＯ＝0の場合： 　メインシステムからＦＤＣへデータを転送する場合で，メインシステムがＦＤＣのデータレジスタにデータをセットしたとき（ＷＲ＝0）にＲＱＭは0となり，ＦＤＣがそのデータを引取ったとき1となります. ＤＩＯ＝1の場合： 　ＦＤＣからメインシステムへデータ転送する場合で，ＦＤＣがデータレジスタにデータをセットしたとき1となり，メインシステムがそのデータを引取ったとき（$\overline{RD}$＝0）0となります.

3－A－3　μＰＤ７６５のコマンド動作

μＰＤ７６５には，ディスクの処理を行なうために１５種のコマンドが用意されており，これらを利用することでディスクの処理を比較的簡単な手順で行なうことができます．次に使用できるコマンドの名称を示します．

- ●ＲＥＡＤ　ＤＡＴＡ
- ●ＷＲＩＴＥ　ＤＡＴＡ
- ●ＷＲＩＴＥ　ＤＥＬＥＴＥＤ　ＤＡＴＡ
- ●ＲＥＡＤ　ＤＩＡＧＮＯＳＴＩＣ
- ●ＲＥＡＤ　ＩＤ
- ●ＷＲＩＴＥ　ＩＤ
- ●ＲＥＡＤ　ＤＥＬＥＴＥＤ　ＤＡＴＡ
- ●ＳＣＡＮ　ＥＱＵＡＬ
- ●ＳＣＡＮ　ＬＯＷ　ＯＲ　ＥＱＵＡＬ
- ●ＳＣＡＮ　ＨＩＧＨ　ＯＲ　ＥＱＵＡＬ
- ●ＳＥＥＫ
- ●ＲＥＣＡＬＩＢＲＡＴＥ
- ●ＳＥＮＳＥ　ＩＮＴＥＲＲＵＰＴ　ＳＴＡＴＵＳ
- ●ＳＰＥＣＩＦＹ
- ●ＳＥＮＳＥ　ＤＥＶＩＣＥ　ＳＴＡＴＵＳ

ＦＤＣは，メインシステムから与えられるコマンドを次の３段階に分けて実行します．

１）コマンド・フェーズ（ＣＰ）

ＦＤＣがアイドル状態にあるときにメインシステムからのコマンドを受け取ると，ＦＤＣはそれに引き続くパラメータを順に受け取ります．ここで，メインシステム側はパラメータを決められたフォーマットに従って送る必要があります．

２）エクセキュージョン・フェーズ（ＥＰ）

コマンド・フェーズで与えられたパラメータに従ってコマンドを実行します．リード・ライト系のコマンドについては，このときにメインシステムとのデータ転送を行います．データの転送はＤＭＡ転送または割り込み処理によって行います．

３）レザルト・フェーズ（ＲＰ）

エクセキュージョン・フェーズが終了するとＦＤＣはメインシステム

側に割り込みをかけて実行の終了を知らせますが（SENSE INTERPUPT STATUS，SENSE DEVICE STATUS，SPECIFYコマンドを除く），ここではそれらコマンドの実行結果をデータレジスタを介して読み出します．メインシステム側は全てのステータス情報を決められた順序に従って読み取る必要があります．

以下にステータス情報（Fig. 3－A－2）と各コマンドのパラメータ・フォーマット（Fig. 3－A－3）を示します．

ビット番号	ステータス名称	略　称	内　　　　容
D7 D6	Interrupt Code	IC	INT要求が何によるかを示します． D7D6 0 0　コマンドの正常終了（NT） 0 1　コマンドの異常終了（AT） 1 0　起動されたコマンドがInvalidであったため，コマンドを実行しなかった事を示します．(IC) 1 1　デバイスの状態遷移があった事を示します．(AI)
D5	Seek End	SE	SEEKまたはRECALIBRATEコマンドによるシーク動作が正常終了または異常終了したときにセットされます．
D4	Equipment Check	EC	デバイスからFault信号を受取ったとき，またはRECALIBRATEコマンド時TRACK0信号が一定時間内に検出できなかったときセットされます．
D3	Not Ready	NR	デバイスがリード／ライト不可能なとき，または片面メディアを使用してSide1にコマンドを出したときセットされます．
D2	Head Address	HD	INT要求時のヘッドの状態を示します． SENSE INTERRUPT STATUSコマンド実行時は常に0となっています．
D1	Unit Select1	US1	INT要求時のデバイス番号を示します．
D0	Unit Select0	US0	

NT：Normal Terminate,　　AT：Abnormal Terminate
IC：Invalid Command,　　AI：Attention Interrupt

Fig. 3－A－2　ステータス情報

レザルトステータス１　（ＳＴ１）

ビット番号	ステータス名称	略　称	内　　　容
D 7	End of Cyljnder	E N	指定された最終セクタを越えてアクセスを継続させようとしたときセットされます.
D 5	Data Error	D E	ディスク上のＩＤまたはデータを読取った際, ＣＲＣエラーを検出するとセットされます. ＩＤ, データの区別はＳＴ２のＤＤビットによります.
D 4	Over Run	O R	あるセクタのデータを処理しているとき, メインシステムのサービスが規定時間内に行われなかったときにセットされます.
D 2	No Data	N D	1 次の４種のコマンド実行時にＩＤＲで指定したセクタがトラック上で検出されなかったときにセットされます. (READ DATA, WRITE DATA, WRITE DELETED DATA, SCAN) 2 READ IDコマンド実行時トラック上にエラーのないＩＤが検出されなかったときにセットされます. 3 READ DIAGNOSTIC コマンド実行時セクタＩＤとＩＤＲの内容が一致しなかったときにセットされます.
D 1	Not Writable	NW	WRITE DATA, WRITE DELETED DATA, WRITE IDコマンドを実行時に, 書込み不可状態を検出するとセットされます.
D 0	Missing Address Mark	M A	1 ディスクのＩＤをアクセスするコマンドで, ＩＭ(Index Mark)を２回検出するまでにＡＭ(Address Mark)が検出されなかったときにセットされます. 2 ディスクのデータを読取るときＤＡＭまたはＤＤＡＭが検出されなかった場合セットされます. この際ＳＴ２のＭＤビットもセットされます.

備考　Ｄ６とＤ３は使用されず常に０です.

レザルトステータス２（ＳＴ２）

ビット番号	ステータス名称	略　称	内　　　　容
Ｄ６	Control Mark	ＣＭ	ＲＥＡＤ ＤＡＴＡまたはＳＣＡＮコマンド実行時にＤＤＡＭの付いたセクタのデータを処理したときセットされます．
Ｄ５	Data Error in Data Field	ＤＤ	ディスクのデータを読取るときＣＲＣエラーを検出するとセットされます．
Ｄ４	No Cylinder	ＮＣ	ＳＴ１のＮＤビットに付帯したステータスで，ＩＤのＣバイトが一致しなかったときセットされます．
Ｄ３	Scan Equal Hit	ＳＨ	ＳＣＡＮコマンドでEqual条件が満足されたときセットされます．
Ｄ２	Scan Not Satisfied	ＳＮ	ＳＣＡＮコマンドで最終セクタまでスキャンしても条件に合うデータが検出されなかったときセットされます．
Ｄ１	Bad Cylinder	ＢＣ	ＳＴ１のＮＤビットに付帯したステータスで，ＩＤのＣバイトが一致せずＦＦ(16)であったときセットされます．
Ｄ０	Missing Address Mark in Data Field	ＭＤ	ディスクのデータを読取るときＤＡＭ，またはＤＤＡＭが検出されなかったときセットされます．

備考　Ｄ７は使用されず常に０です．

レザルトステータス３（ＳＴ３）

ビット番号	ステータス名称	略称	内　　　　容
Ｄ７	Fault	ＦＴ	デバイスからのFault信号の状態
Ｄ６	Write Protected	ＷＰ	デバイスからのWrite Protected信号の状態
Ｄ５	Ready	ＲＹ	デバイスからのReady信号の状態
Ｄ４	Track 0	Ｔ０	デバイスからのTrack 0信号の状態
Ｄ３	Two Side	ＴＳ	デバイスからのTwo Side信号の状態
Ｄ２	Head Address	ＨＤ	デバイスへのSide Select信号の状態
Ｄ１	Unit Select 1	ＵＳ１	デバイスへのUnit Select 1信号の状態
Ｄ０	Unit Select 0	ＵＳ０	デバイスへのUnit Select 0信号の状態

コマンド説明に用いられる略称

次に示す略称が，以降のコマンドの説明で使用されています．

略　　　　称	意　　　　味
R／W (Read／Write)	リード(R)または(W)信号がアクティブになることを示します．
Ao	コントロール信号Aoの状態を示します．
D_7 －D_0 (Data Bus)	8ビットデータ・バスとの対応を示します．
MT (Multitrack)	マルチトラック操作を指定します．MTが1ならマルチトラック操作が許可され，0なら禁止されます．
MF (MFM Mode)	MFMモードかFMモードかの指定をします．1のときMFMモード，0のときFMモードを指定します．
SK (Skip)	Deleted Data Address Markをスキップすることを指定します．
HD (Head)	物理的ヘッド番号0または1を指定します．
USI, US0 (Unit Select)	ドライブユニット番号を示します．
C (Cylinder Address)	メディアのシリンダ番号0～76を示します．
H (Head Address)	論理的ヘッド番号を示します．
R (Record(Sector)Address)	セクタ番号を示します．
N (Record(Sector)Length)	1セクタ内のデータ長を示します．
EOT (End of Track)	あるトラック上の最終セクタ番号を示します．
GPL (Gap Length)	VFO SYNCを除いたGap 3の長さ（バイト数）を示します．
ST0, ST1, ST2 (Status)	レザルトステータス情報をストアするレジスタ
ST3 (Status)	ドライブの状態を示す情報をストアするレジスタ

略　　　称	意　　　味
SC (Sector)	トラック当りのセクタ数を示します.
D (Data)	データフィールドに書込むデータのパターンを示します
STP (Step)	スキャンコマンドを実行する場合，あるセクタが終了したら次のセクタへ進むか，１つあけた次のセクタへ進むかの指定をします．STPが1のときは続けて次のセクタをスキャンし，２のときは1つセクタをあけて次のスキャンを行います.
NCN (New Cylinder Number)	シーク操作の結果として新しく選択されるべきシリンダ番号を示します.
PCN (Present Cylinder Number)	SENSE INTERRUPT STATUSコマンドの終了時点におけるシリンダ番号を示します.
SRT (Step Rate Time)	SPECIFYコマンドによって定義されるステップレイト時間を示します.
HUT (Head Unload Time)	SPECIFYコマンドによって定義されるヘッド・アンロード時間を示します.
HLT (Head Load Time)	SPECIFYコマンドによって定義されるヘッド・ロード時間を示します.
ND (Non DMA Mode)	Non DMAモードを示し，SPECIFYコマンドによって指定されます.

各コマンドのコマンドフェーズにおける書込み順序，レザルトフェーズにおける読取り順序は，この一覧に記述されている順序に従って下さい.

C ：Command Phase
E ：Execution Phase
R ：Result Phase

Fig. 5—A—3 コマンド一覧表

コマンド		R/W A0	D7	D6	D5	D4	D3	D2	D1	D0	備考
READ DATA	C	W1	MT	MF	SK	0	0	1	1	0	SK :SKip DDAM MT:マルチトラック, MF:MFMモード
			×	×	×	×	×	HD	US1	US0	HD：ヘッド番号, US：デバイス番号
			◀— C —▶								実行開始セクタのID情報
			◀— H —▶								
			◀— R —▶								
			◀— N —▶								
			◀— EOT —▶								トラック上の最終セクタ番号
			◀— GPL —▶								Gap3の長さ(VFO SYNCを含まず)
		W1	◀— DTA —▶								処理すべきセクタ当りのデータ長
	E										データ転送
	R	R1	◀— ST 0 —▶								実行終了時のステータス0
			◀— ST 1 —▶								〃 ステータス1
			◀— ST 2 —▶								〃 ステータス2
			◀— C —▶								実行終了セクタのID情報
			◀— H —▶								
			◀— R —▶								
		R1	◀— N —▶								
READ DELETED DATA	C	W1	MT	MF	SK	0	1	1	0	0	SK：Skip DAM
			×	×	×	×	×	HD	US1	US0	READ DATAと同じ
			◀— C —▶								
			◀— H —▶								
			◀— R —▶								
			◀— N —▶								
			◀— EOT —▶								
			◀— GPL —▶								
		W1	◀— DTL —▶								
	E										データ転送
	R	R1	◀— ST 0 —▶								READ DATAと同じ
			◀— ST 1 —▶								
			◀— ST 2 —▶								
			◀— C —▶								
			◀— H —▶								
			◀— R —▶								
		R1	◀— N —▶								

コマンド	R/WA0 D7 ―――― D0	備考
READ ID		
C	W1 0 MF 0 0 1 0 1 0	
	W1 × × × × × HD US1 US0	トラック上の最初に読んだエラーのないID情報をストアする
E		
R	R1 ← ST0 →	READ DATAと同じ
	← ST1 →	
	← ST2 →	
	← C →	E−Phaseで読んだID情報
	← H →	
	← R →	
	R1 ← N →	
WRITE ID		
C	W1 0 MF 0 0 1 1 0 1	
	× × × × × HD US1 US0	
	← N →	データ長／セクタ
	← SC →	セクタ数／トラック
	← GPL →	Gap3の長さ（VFO SYNCを含まず）
	W1 ← D →	データ領域に書込むデータパターン
E		トラック上のセクタ数分のID情報をメインシステムより転送する
R	R1 ← ST0 →	READ DATAと同じ（ただし、C, H, Rは無意味）
	← ST1 →	
	← ST2 →	
	← C →	
	← H →	
	← R →	
	R1 ← N →	

コマンド		R/W	A0	D7 — D0	備考
WRITE DATA	C	W	1	MT MF 0 0 0 1 0 1	READ DATAと同じ
				× × × × × HD US1 US0	
				◀— C —▶	
				◀— H —▶	
				◀— R —▶	
				◀— N —▶	
				◀— EOT —▶	
				◀— GPL —▶	
		W	1	◀— DTL —▶	
	E				データ転送
	R	R	1	◀— ST 0 —▶	READ DATAと同じ
				◀— ST 1 —▶	
				◀— ST 2 —▶	
				◀— C —▶	
				◀— H —▶	
				◀— R —▶	
		R	1	◀— N —▶	
WRITE DELETED DATA	C	W	1	MT MF 0 0 1 0 0 1	READ DATAと同じ
				× × × × × HD US1 US0	
				◀— C —▶	
				◀— H —▶	
				◀— R —▶	
				◀— N —▶	
				◀— EOT —▶	
				◀— GPL —▶	
		W	1	◀— DTL —▶	
	E				データ転送
	R	R	1	◀— ST 0 —▶	READ DATAと同じ
				◀— ST 1 —▶	
				◀— ST 2 —▶	
				◀— C —▶	
				◀— H —▶	
				◀— R —▶	
		R	1	◀— N —▶	

コマンド		R/W	A0	D7							D0	備考
READ DIAGNOSTIC	C	W	1	MT	MF	0	0	0	0	1	0	READ DATAと同じ
				×	×	×	×	×	HD	US1	US0	
				←			C				→	
				←			H				→	
				←			R				→	
				←			N				→	
				←			EOT				→	
				←			GPL				→	
		W	1	←			DTL				→	
	E											データ転送
	R	R	1	←			ST0				→	READ DATAと同じ
				←			ST1				→	
				←			ST2				→	
				←			C				→	
				←			H				→	
				←			R				→	
		R	1	←			N				→	
SCAN EQUAL	C	W	1	MT	MF	SK	1	0	0	0	1	READ DATAと同じ
				×	×	×	×	×	HD	US1	US0	
				←			C				→	
				←			H				→	
				←			R				→	
				←			N				→	
				←			EOT				→	
				←			GPL				→	
		W	1	←			STP				→	比較すべきセクタのセクタ間隔 1or2
	E											データ転送
	R	R	1	←			ST0				→	READ DATAと同じ
				←			ST1				→	
				←			ST2				→	
				←			C				→	
				←			H				→	
				←			R				→	
		R	1	←			N				→	

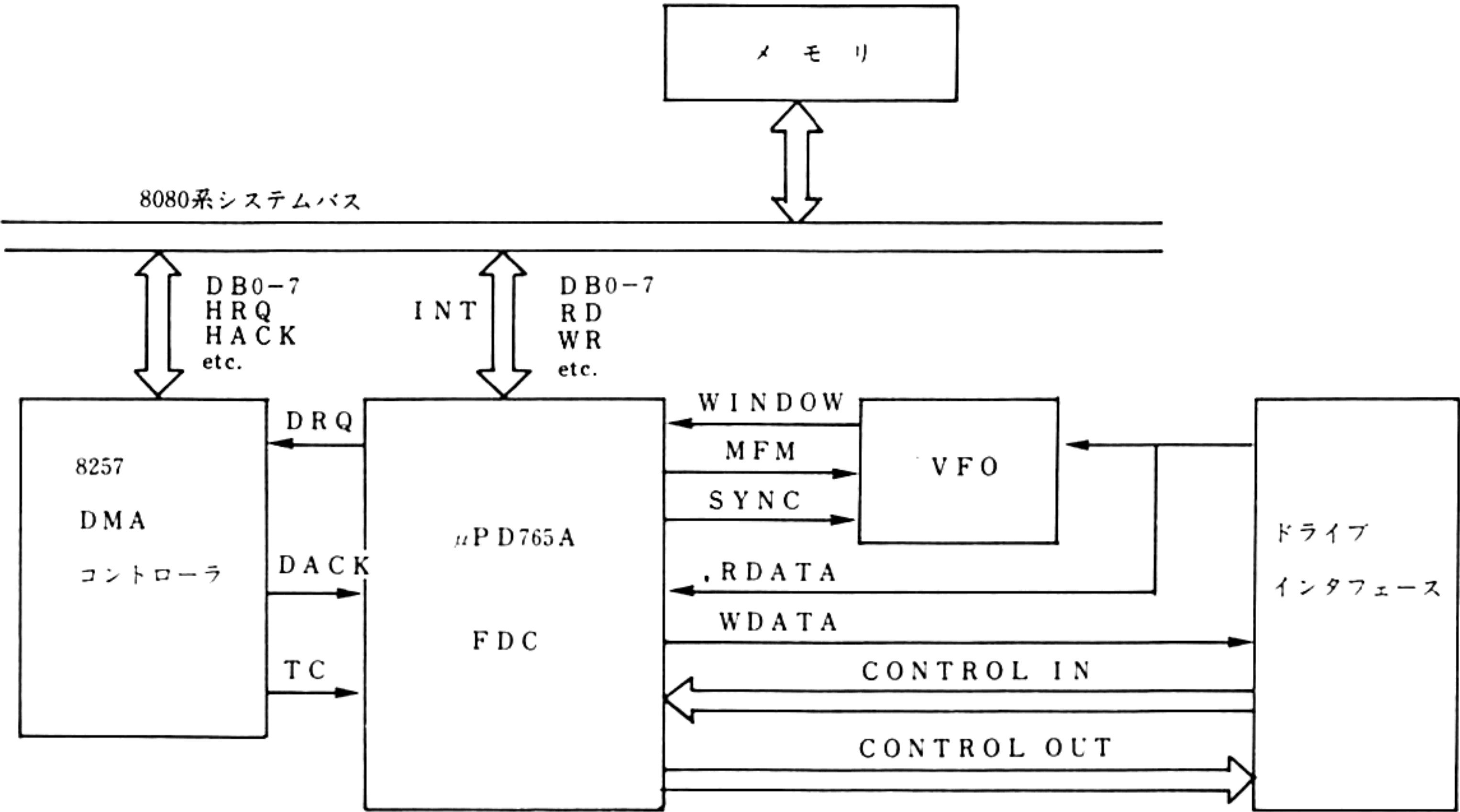
コマンド		R/W A0	D7 ——————— D0	備考
SCAN LOW OR EQUAL	C	W 1	MT MF SK 1 1 0 0 1	READ DATAと同じ
			× × × × × HD US1 US0	
			◀— C —▶	
			◀— H —▶	
			◀— R —▶	
			◀— N —▶	
			◀— EOT —▶	
			◀— GPL —▶	
		W 1	◀— STP —▶	比較すべきセクタのセクタ間隔 1or2
	E			データ転送
	R	R 1	◀— ST 0 —▶	READ DATAと同じ
			◀— ST 1 —▶	
			◀— ST 2 —▶	
			◀— C —▶	
			◀— H —▶	
			◀— R —▶	
		R 1	◀— N —▶	
SCAN HIGH OR EQUAL	C	W 1	MT MF SK 1 1 1 0 1	READ DATAと同じ
			× × × × × HD US1 US0	
			◀— C —▶	
			◀— H —▶	
			◀— R —▶	
			◀— N —▶	
			◀— EOT —▶	
			◀— GPL —▶	
		W 1	◀— STP —▶	比較すべきセクタのセクタ間隔 1or2
	E			データ転送
	R	R 1	◀— ST 0 —▶	READ DATAと同じ
			◀— ST 1 —▶	
			◀— ST 2 —▶	
			◀— C —▶	
			◀— H —▶	
			◀— R —▶	
		R 1	◀— N —▶	

コマンド	R/WA0 D0 ——— D0								備考
SEEK	W1	0	0	0	0	1	1	1	1	
	W1	×	×	×	×	×	HD	US1	US0	
		← NCN →								新シリンダ番号
										シーク動作
RECALIBRATE C	W1	0	0	0	0	0	1	1	1	
	W1	×	×	×	×	×	0	US1	US0	
E										リキャリブレイト動作
SENSE INT STATUS C	W1	0	0	0	0	1	0	0	0	
R	R1	← ST0 →								
	R1	← PCN →								コマンド終了時のシリンダ番号
SENSE DEVICE STATUS C	W1	0	0	0	0	0	1	0	0	
	W1	×	×	×	×	×	HD	US1	US0	
R	R1	← ST3 →								デバイスの状態
SPECIFY C	W1	0	0	0	0	0	0	1	1	Step Rate Time, Head Unload Time
	W1	← →								
	W1	← HLT →							ND	Head Load Time, Non DMA Mode
Invalid C	W1	← その他のコード →								
R	R1	← ST0 →								ST0＝80 (16)

３－Ａ－４　実際のインタフェース

ＰＣ－８８０１では本体後部のスロットバス３・４にＤＭＡ転送方式のディスクドライブ用のインタフェース・ボードを装着することが可能です．Ｉ／Ｏアドレスは８インチ用と５インチ用別々に，次の様に割り当てられています．

	8″	5″
インタフェースセレクト	Ｆ３Ｈ	
インタフェースチェック	Ｆ４Ｈ	Ｆ８Ｈ
マージンコントロール	Ｆ５Ｈ	Ｆ９Ｈ
ＦＤＣステータスレジスタ	Ｆ６Ｈ	ＦＡＨ
ＦＤＣデータレジスタ	Ｆ７Ｈ	ＦＢＨ

また，ＰＣ－８８０１ではディスクドライブとのデータ転送にＤＭＡ方式を採用しており，そのＬＳＩにはμＰＤ８２５７が使用されています．以下に，ＤＭＡＣ（μＰＤ８２５７）を含めたブロック図を示します．

４６８ＣＨ　ファイル・ディスプリクタの処理を行う．

アドレス	４６８ＣＨ
機　能	ファイル・ディスプリクタの処理を行う．
レジスタ	Ａ，Ｆ，Ｂ，Ｃ，Ｄ，Ｅ，Ｈ，Ｌ

解　説　ＨＬレジスタで示されるアドレスより格納されている文字列をファイルディスプリクタとして処理し，デバイス・ナンバとファイル名を得るサブルーチンです．結果は以下の様に得られます．

　　　Ａレジスタ・Ｄレジスタ………デバイス・ナンバ（１－９を参照）．

　　　ＥＣ８ＦＨ～ＥＣ９７Ｈ………ファイル名（拡張子を含む）．

　このとき，ＨＬレジスタの示す文字列の先頭は　”　（ダブルクォーテーション）の必要があり，文字列の最後は　”　もしくは００Ｈの必要があります．また，文字列の長さが０であった場合や，ファイルディスプリタとして不都合な場合は Bad file name エラーが発生します．

　デバイス名を省略した場合のデバイス・ナンバのデフォルト値は，００Ｈ（ディスクドライブ1）です．

サンプル

```
                ;
                ; --- get device number,file name and expansion
                ;
                ;       sample of subroutine from 468CH
                ;
                        ORG   0B900H
                ;
B900 2176B9     DSPDES:LD    HL,FILEDS          ; input file descriptor
B903 CD6DB9            CALL  DSPMES
B906 CDC85F            CALL  5FC8H              ; line input from keyboard
B909 D8                RET   C                  ; stop key return
                ;
B90A 3622              LD    (HL),'"'
B90C CD8C46            CALL  468CH              ; processing of file descriptor
                ;
B90F 218BB9            LD    HL,DEVNUM          ; print device number
B912 CD6DB9            CALL  DSPMES
B915 7A                LD    A,D
B916 E6F0              AND   0F0H               ; mask lower bits
B918 0F                RRCA
B919 0F                RRCA
B91A 0F                RRCA
B91B 0F                RRCA
B91C C630              ADD   A,'0'              ; convert to ascii
B91E FE3A              CP    '9'+1
B920 3802              JR    C,CONVJ1
                ;
B922 C607              ADD   A,7                ; alphabet
B924 CD0D3E     CONVJ1:CALL  3E0DH
B927 7A                LD    A,D
B928 E60F              AND   0FH                ; mask upper bits
B92A C630              ADD   A,'0'
B92C FE3A              CP    '9'+1
B92E 3802              JR    C,CONVJ2
```

```
                    ;
B930 C607 .           ADD  A,7
B932 CD0D3E    CONVJ2:CALL 3E0DH
B935 21AAB9           LD   HL,HCR          ; print 'H' and line feed
B938 CD6DB9           CALL DSPMES
                    ;
B93B 2195B9           LD   HL,FILENM
B93E CD6DB9           CALL DSPMES
B941 218FEC           LD   HL,0EC8FH       ; top of filename buffer
B944 0606             LD   B,6
B946 7E        DSPFIL:LD   A,(HL)
B947 CD0D3E           CALL 3E0DH
B94A 23               INC  HL
B94B 10F9             DJNZ DSPFIL
                    ;
B94D 21ABB9           LD   HL,HCR+1        ; line feed
B950 CD6DB9           CALL DSPMES
                    ;
B953 219FB9           LD   HL,EXPANS
B956 CD6DB9           CALL DSPMES
B959 2195EC           LD   HL,0EC8FH+6     ; top of expansion
B95C 0603             LD   B,3
B95E 7E        DSPEXP:LD   A,(HL)
B95F CD0D3E           CALL 3E0DH
B962 23               INC  HL
B963 10F9             DJNZ DSPEXP
                    ;
B965 21ABB9           LD   HL,HCR+1
B968 CD6DB9           CALL DSPMES
                    ;
B96B 1893             JR   DSPDES
                    ;
B96D 7E        DSPMES:LD   A,(HL)          ; display message subroutine
B96E A7               AND  A
B96F C8               RET  Z
B970 CD0D3E           CALL 3E0DH
B973 23               INC  HL
B974 18F7             JR   DSPMES
                    ;
B976 0D0A4669 FILEDS:DEFM 0DH,0AH,'File descriptor ? ',0
B97A 6C652064
B97E 65736372
B982 6970746F
B986 72203F20
B98A 00
                    ;
B98B 0D0A4445 DEVNUM:DEFM 0DH,0AH,'DEVICE:',0
B98F 56494345
B993 3A00
B995 46494C45 FILENM:DEFM 'FILENAME:',0
B999 4E414D45
B99D 3A00
B99F 45585041 EXPANS:DEFM 'EXPANSION:',0
B9A3 4E53494F
B9A7 4E3A00
                    ;
B9AA 480D0A00 HCR:    DEFM 'H',0DH,0AH,0
                    ;
B9AE                  END
```

```
a=&hb900:call a


File descriptor ? 1:backupn88

DEVICE:00H
FILENAME:backup
EXPANSION:n88
```

```
File descriptor ? scrn:output

DEVICE:F8H
FILENAME:output
EXPANSION:

File descriptor ?
Ok
```

４６Ｆ８Ｈ　ファイルバッファのアドレスを求める.

アドレス	４６Ｆ８Ｈ
機　能	ファイルバッファのアドレスを求める.
レジスタ	Ａ，Ｆ，Ｄ，Ｅ，Ｈ，Ｌ

解　説　アキュムレータの値で指定されるファイルの，ファイルバッファの先頭アドレスを求め，値をＨＬレジスタペアに与えます．また，その際，以下の様に指定ファイルに関する情報を得ることができます．

ＣＹフラグ………指定ファイルがディスクファイル以外のものであった場合にセットされます．

Ｚフラグ　………指定ファイルがオープン状態であるかどうかを示します．クローズ状態のときにセットされます．

４７４２Ｈ　ディスクの諸元をセットする．

アドレス　４７４２Ｈ

機　能　ディスクの諸元をセットする．

レジスタ

解　説　アキュムレータの値＋１で示されるドライブに関する情報をワークエリアにコピーします．このときセットされるデータは以下のとおりです．

ＥＣ８５Ｈ：ドライブ・ナンバー１．
ＥＣＡ５Ｈ：最大トラック・ナンバ．
ＥＣＡ６Ｈ：１トラックあたりのセクタ数．
ＥＣＡ７Ｈ：サーフェス・フラグ（００Ｈ：片面，０１Ｈ：両面）．
ＥＣＡ８Ｈ：１トラックあたりのクラスタ数．
ＥＣＡ９Ｈ：１ボリュームあたりのクラスタ数．
ＥＣＡＡＨ：ディレクトリのあるトラック・ナンバ．
ＥＣＡＢＨ：１クラスタあたりのセクタ数．
ＥＣＡＣＨ：ＦＡＴの先頭セクタ・ナンバ．
ＥＣＡＤＨ：ＦＡＴの終了セクタ・ナンバ．
ＥＣＡＥＨ：ＦＡＴの数．
ＥＣＡＦＨ：ディスク属性の入っているセクタ・ナンバ．

なお，ディスクをアクセスするルーチンのなかには，これらのワークエリアの値を参照するものが多数存在するため，ドライブ・ナンバを読み込んだ際にはこのサブルーチンを実行させておく必要があります．

サンプル

```
               ;
               ; --- output DSKF(X,0-10) function
               ;
               ;       sample of subroutine from 4742H
               ;
                      ORG   0B900H
               ;
B900 2188B9    PRDSKF:LD    HL,DRIVE          ; input drive number
B903 CD69B9           CALL  DSPMES
B906 CDC85F           CALL  5FC8H             ; subroutine line input
B909 D8               RET   C                 ; stop key,return
               ;
B90A 23               INC   HL
B90B CDBC26           CALL  26BCH             ; convert to binary
B90E CDA021           CALL  21A0H             ; convert FAC to integer
               ;
B911 2D               DEC   L
B912 FA00B9           JP    M,PRDSKF          ; drive number minus or 0
```

```
                  ;
B915 3A7DEC               LD    A,(0EC7DH)          ; maximum drive number
B918 3D                   DEC   A
B919 BD                   CP    L
B91A 38E4                 JR    C,PRDSKF            ; over max
                  ;
B91C 7D                   LD    A,L
B91D CD4247               CALL  4742H               ; set disk identification
                  ;
B920 21A5EC               LD    HL,0ECA5H           ; top of DSKF table
B923 0E00                 LD    C,0                 ; index clear
B925 0600                 LD    B,0
                  ;
B927 E5           LOOP:   PUSH  HL                  ; message output
B928 2172B9               LD    HL,MESTOP
B92B 09                   ADD   HL,BC
B92C 09                   ADD   HL,BC
B92D 5E                   LD    E,(HL)
B92E 23                   INC   HL
B92F 56                   LD    D,(HL)
B930 EB                   EX    DE,HL
B931 CD69B9               CALL  DSPMES
B934 E1                   POP   HL
                  ;
B935 79                   LD    A,C
B936 FE02                 CP    2
B938 281E                 JR    Z,SIDE              ; side output
                  ;
B93A 7E                   LD    A,(HL)              ; get DSKF data
B93B E5                   PUSH  HL
B93C C5                   PUSH  BC
B93D 2600                 LD    H,0
B93F 6F                   LD    L,A
B940 CDC228               CALL  28C2H               ; output integer
B943 C1                   POP   BC
B944 E1                   POP   HL
                  ;
B945 3E0D         CRLF:   LD    A,0DH               ; CR,LF output
B947 CD0D3E               CALL  3E0DH
B94A 3E0A                 LD    A,0AH
B94C CD0D3E               CALL  3E0DH
                  ;
B94F 23                   INC   HL                  ; next data
B950 0C                   INC   C
B951 79                   LD    A,C
B952 FE0B                 CP    11
B954 20D1                 JR    NZ,LOOP
                  ;
B956 18A8                 JR    PRDSKF              ; all data output
                  ;
B958 7E           SIDE:   LD    A,(HL)
B959 E5                   PUSH  HL
B95A 219AB9               LD    HL,SINGLE           ; single surface
B95D B7                   OR    A
B95E 2803                 JR    Z,SIDES
B960 21AAB9               LD    HL,DOUBLE           ; double surface
B963 CD69B9       SIDES:  CALL  DSPMES
B966 E1                   POP   HL
B967 18DC                 JR    CRLF                ; goto main
                  ;
                  ; subroutine display message
                  ;
B969 7E           DSPMES:LD     A,(HL)
B96A A7                   AND   A
B96B C8                   RET   Z                   ; 00H,return
B96C CD0D3E               CALL  3E0DH
B96F 23                   INC   HL
B970 18F7                 JR    DSPMES
                  ;
                  ; address table area
                  ;
B972 BAB9         MESTOP:DEFW   MAXTRK
B974 D2B9                 DEFW  SECTRK
B976 E8B9                 DEFW  SIDFLG
```

```
B978 F0B9               DEFW CLSTRK
B97A 07BA               DEFW CLSVOL
B97C 1FBA               DEFW DIRTRK
B97E 2CBA               DEFW SECCLS
B980 44BA               DEFW FATTOP
B982 59BA               DEFW FATEND
B984 6EBA               DEFW FATNUM
B986 7FBA               DEFW IDSEC
                 ;
                 ; message area
                 ;
B988 0D0A4472 DRIVE: DEFM 0DH,0AH,'Drive number ? ',0
B98C 69766520
B990 6E756D62
B994 6572203F
B998 2000
                 ;
B99A 53494E47 SINGLE:DEFM 'SINGLE SURFACE.',0
B99E 4C452053
B9A2 55524641
B9A6 43452E00
B9AA 444F5542 DOUBLE:DEFM 'DOUBLE SURFACE.',0
B9AE 4C452053
B9B2 55524641
B9B6 43452E00
                 ;
B9BA 4D617869 MAXTRK:DEFM 'Maximum track number...',0
B9BE 6D756D20
B9C2 74726163
B9C6 6B206E75
B9CA 6D626572
B9CE 2E2E2E00
B9D2 53656374 SECTRK:DEFM 'Sector par a track...',0
B9D6 6F722070
B9DA 61722061
B9DE 20747261
B9E2 636B2E2E
B9E6 2E00
B9E8 53696465 SIDFLG:DEFM 'Side...',0
B9EC 2E2E2E00
B9F0 436C7573 CLSTRK:DEFM 'Cluster par a track...',0
B9F4 74657220
B9F8 70617220
B9FC 61207472
BA00 61636B2E
BA04 2E2E00
BA07 436C7573 CLSVOL:DEFM 'Cluster par a volume...',0
BA0B 74657220
BA0F 70617220
BA13 6120766F
BA17 6C756D65
BA1B 2E2E2E00
BA1F 44495220 DIRTRK:DEFM 'DIR track...',0
BA23 74726163
BA27 6B2E2E2E
BA2B 00
BA2C 53656374 SECCLS:DEFM 'Sector par a cluster...',0
BA30 6F722070
BA34 61722061
BA38 20636C75
BA3C 73746572
BA40 2E2E2E00
BA44 546F7020 FATTOP:DEFM 'Top sector of FAT...',0
BA48 73656374
BA4C 6F72206F
BA50 66204641
BA54 542E2E2E
BA58 00
BA59 456E6420 FATEND:DEFM 'End sector of FAT...',0
BA5D 73656374
BA61 6F72206F
BA65 66204641
BA69 542E2E2E
BA6D 00
```

```
BA6E 4E756D62 FATNUM:DEFM ´Number of FAT...´,0
BA72 6572206F
BA76 66204641
BA7A 542E2E2E
BA7E 00
BA7F 49442073 IDSEC: DEFM ´ID sector...´,0
BA83 6563746F
BA87 722E2E2E
BA8B 00
               ;
BA8C                    END
```

```
a=&hb900:call a

Drive number ? 1
Maximum track number...39
Sector par a track...16
Side...DOUBLE SURFACE.
Cluster par a track...2
Cluster par a volume...160
DIR track...18
Sector par a cluster...8
Top sector of FAT...14
End sector of FAT...16
Number of FAT...3
ID sector...13

Drive number ?
Ok
```

47F6H ファイルをオープンする.

アドレス 47F6H

機　能 指定ファイルをオープンする.

レジスタ A, F, B, C, D, E

解　説 Aレジスタで指定されるファイルをDレジスタで指定されるデバイス上にオープンします. そのときに必要なパラメータは次のようになります.

Aレジスタ ………ファイル・ナンバ.

Dレジスタ ………デバイス・ナンバ (1－9を参照).

Eレジスタ ………ファイルモード (1－9を参照).

EC8FH～EC97H

………ファイル名.

また, このルーチン中にテキスト解釈が含まれているため, テキストポインタ (HLレジスタ) に00Hあるいは3AHをポイントさせる必要があります.

指定したファイルがすでに開いていた場合には, File already OPENエラーが発生します.

サンプル

```
              ;
              ; --- kill disk file
              ;
              ;       sample of subroutine from 47F6H
              ;
                      ORG   0B900H
              ;
B900 2149B9   KILLFL:LD    HL,FILENM          ; input file name
B903 CD40B9           CALL DSPMES
B906 CDC85F           CALL 5FC8H              ; line input subroutine
B909 D8               RET  C                  ; stop key return
              ;
B90A 3622             LD   (HL),'"'
B90C CD8C46           CALL 468CH              ; processing of file descriptor
B90F A7               AND  A
B910 FA00B9           JP   M,KILLFL           ; non disk file
              ;
B913 F5               PUSH AF
B914 1E10             LD   E,10H              ; kill mode
B916 AF               XOR  A                  ; file#=0
B917 CDF647           CALL 47F6H              ; file open with kill mode
              ;
B91A AF               XOR  A
B91B CD1D48           CALL 481DH              ; file close
              ;
B91E F1               POP  AF
B91F CDEA97           CALL 97EAH              ; remove device
              ;
B922 2158B9           LD   HL,KILMES          ; message of kill
B925 CD40B9           CALL DSPMES
B928 218FEC           LD   HL,0EC8FH          ; top of name buffer
```

```
B92B 0609               LD   B,9
B92D 7E        DLOOP:   LD   A,(HL)
B92E CD0D3E             CALL 3E0DH
B931 23                 INC  HL
B932 10F9               DJNZ DLOOP
               ;
B934 3E0D               LD   A,0DH              ; line feed
B936 CD0D3E             CALL 3E0DH
B939 3E0A               LD   A,0AH
B93B CD0D3E             CALL 3E0DH
               ;
B93E 18C0               JR   KILLFL
               ;
B940 7E        DSPMES:LD     A,(HL)             ; subroutine display message
B941 A7                 AND  A
B942 C8                 RET  Z
B943 CD0D3E             CALL 3E0DH
B946 23                 INC  HL
B947 18F7               JR   DSPMES
               ;
B949 0D0A4669 FILENM:DEFM 0DH,0AH,'File name ? ',0
B94D 6C65206E
B951 616D6520
B955 3F2000
B958 4B494C4C KILMES:DEFM 'KILLED FILE:',0
B95C 45442046
B960 494C453A
B964 00
               ;
B965                    END
```

```
files
backup.n88 2    format.n88 2    setinf.n88 1    xfiles.n88 1    sysgen.    1

Ok
a=&hb900:call a

File name ? format.n88
KILLED FILE:formatn88

File name ?
Ok
files
backup.n88 2    setinf.n88 1    xfiles.n88 1    sysgen.    1
Ok
```

481DH　ファイルのクローズ.

アドレス	481DH
機　能	指定ファイルをクローズする.
レジスタ	A, F, D, E

解　説　アキュムレータで指定されるファイルのクローズを行ないます. 指定されたファイルがすでにクローズされていた場合には, エラーにはならず, そのまま処理を行なわずに戻ります.

サンプル

```
               ;
               ; --- close all file
               ;
               ;       sample of subroutine from 481DH
               ;
                      ORG   0B900H
               ;
B900 3A7EEC    CLOSEF:LD    A,(0EC7EH)          ; maximum file number
B903 4F               LD    C,A
               ;
B904 79        CLOSEL:LD    A,C
B905 CD1D48           CALL  481DH               ; close file of #ACC
               ;
B908 0D               DEC   C
B909 F204B9           JP    P,CLOSEL
               ;
B90C 2118B9           LD    HL,CLOSMS           ; message of close
B90F 7E        CLOSML:LD    A,(HL)
B910 A7               AND   A
B911 C8               RET   Z
B912 CD0D3E           CALL  3E0DH
B915 23               INC   HL
B916 18F7             JR    CLOSML
               ;
B918 0D0A416C  CLOSMS:DEFM  0DH,0AH,'All file closed.',0
B91C 6C206669
B920 6C652063
B924 6C6F7365
B928 642E00
               ;
B92B                  END
```

```
open "dummy1" as #1:open "dummy2" as #2
Ok
a=&hb900:call a

All file closed.
Ok
close
Ok
```

４Ｂ５４Ｈ　シーケンシャルファイルへの出力．

アドレス　４Ｂ５４Ｈ

機　能　シーケンシャルファイルへの１バイトの書き込みを行なう．

レジスタ

解　説　アキュムレータに格納されている１バイトのデータをシーケンシャルファイルに書き込んで戻ります．このとき，ＥＣ８８Ｈ，８９Ｈ両番地には指定ファイルのバッファの先頭アドレスがセットされていなければなりません．また，ＥＣ８８Ｈ，８９Ｈ両番地の値が００００Ｈ以外の場合には，００１８Ｈ番地からのサブルーチンも，このサブルーチンと同様の働きをします．

サンプル

```
               ;
               ; --- create sequential file
               ;
               ;        sample of subroutine from 4B54H
               ;
                      ORG  0B900H
               ;
B900 2173B9    CRSQFL:LD   HL,FILENM          ; input file name
B903 CD6AB9           CALL DSPMES
B906 CDC85F           CALL 5FC8H              ; line input from keyboard
B909 D8               RET  C                  ; stop key return
               ;
B90A 3622             LD   (HL),'"'
B90C CD8C46           CALL 468CH              ; processing of file descriptor
               ;
B90F 3E01             LD   A,1                ; file#=1
B911 1E02             LD   E,2                ; output mode
B913 CDF647           CALL 47F6H              ; file open
               ;
B916 2182B9    DATAIN:LD   HL,INPDAT          ; input data
B919 CD6AB9           CALL DSPMES
               ;
B91C CD8335    DATAIL:CALL 3583H              ; input 1 byte data from keyboard
B91F FE0D             CP   0DH                ; line feed
B921 280A             JR   Z,CODE0D
               ;
B923 FE03             CP   03H                ; STOP key
B925 2836             JR   Z,ESCAPE
               ;
B927 FE20             CP   ' '
B929 38F1             JR   C,DATAIL           ; ignore control code
               ;
B92B 181D             JR   MAINCH
               ;
B92D F5        CODE0D:PUSH AF
B92E 3E01             LD   A,1                ; file#=1
B930 CD3547           CALL 4735H              ; set address of buffer
B933 F1               POP  AF
B934 CD544B           CALL 4B54H              ; output to file
B937 3E0A             LD   A,0AH
B939 CD544B           CALL 4B54H
               ;
B93C 210000           LD   HL,0
B93F 2288EC           LD   (0EC88H),HL        ; clear buffer address
```

```
B942 3E0D            LD   A,0DH               ; line feed on screen
B944 DF              RST  18H
B945 3E0A            LD   A,0AH
B947 DF              RST  18H
B948 18CC            JR   DATAIN
                ;
B94A F5         MAINCH:PUSH AF
B94B 3E01            LD   A,1
B94D CD3547          CALL 4735H
B950 F1              POP  AF
B951 CD544B          CALL 4B54H
                ;
B954 210000          LD   HL,0
B957 2288EC          LD   (0EC88H),HL
B95A DF              RST  18H                 ; echo back
B95B 18BF            JR   DATAIL
                ;
B95D 3E01       ESCAPE:LD   A,1               ; file#=1
B95F CD1D48          CALL 481DH               ; file close
                ;
B962 3E0D            LD   A,0DH               ; line feed
B964 DF              RST  18H
B965 3E0A            LD   A,0AH
B967 DF              RST  18H
                ;
B968 1896            JR   CRSQFL
                ;
B96A 7E         DSPMES:LD   A,(HL)            ; display message subroutine
B96B A7              AND  A
B96C C8              RET  Z
B96D CD0D3E          CALL 3E0DH
B970 23              INC  HL
B971 18F7            JR   DSPMES
                ;
B973 0D0A4669 FILENM:DEFM 0DH,0AH,'File name ? ',0
B977 6C65206E
B97B 616D6520
B97F 3F2000
B982 44617461 INPDAT:DEFM 'Data(end:stop):',0
B986 28656E64
B98A 3A73746F
B98E 70293A00
                ;
B992                 END
```

```
a=&hb900:call a

File name ? dummy5
Data(end:stop):This message is for the sample !!
Data(end:stop):

File name ?
Ok
open "dummy5" for input as #1
Ok
while not eof(1):line input #1,a$:print a$:wend
This message is for the sample !!
Ok
close
Ok
```

４Ｂ７ＢＨ　シーケンシャルファイルからの入力．

アドレス　４Ｂ７ＢＨ

機　　能　シーケンシャルファイルからの１バイトの読み込みを行なう．

レジスタ　Ａ，Ｆ

解　　説　シーケンシャルファイルから１バイトのデータを読み込みを行ない，値をアキュムレータに格納して戻ります．読み込むデータがファイルにない場合には，ＣＹフラグをセットして戻ります．このとき，ＥＣ８８Ｈ，８９Ｈ両番地には指定ファイルのバッファの先頭アドレスがセットされていなければなりません．

サンプル

```
                ;
                ; --- get top and end address of machine file
                ;
                ;       sample of subroutine from 4B7BH
                ;
                        ORG  0B900H
                ;
B900 218BB9     GETADR:LD   HL,FILENM           ; input file name
B903 CD7BB9            CALL DSPMES
B906 CDC85F            CALL 5FC8H               ; line input subroutine
B909 D8                RET  C                   ; stop key return
                ;
B90A 3622              LD   (HL),'"'
B90C CD8C46            CALL 468CH               ; processing file desciptor
                ;
B90F 3E00              LD   A,0                 ; file#=0
B911 1E01              LD   E,1                 ; input mode
B913 CDF647            CALL 47F6H               ; file open
                ;
B916 CD7B4B            CALL 4B7BH               ; input 1 byte data from file
B919 5F                LD   E,A                 ; get top address
B91A CD7B4B            CALL 4B7BH
B91D 57                LD   D,A
B91E CD7B4B            CALL 4B7BH
B921 6F                LD   L,A                 ; get end address+1
B922 CD7B4B            CALL 4B7BH
B925 67                LD   H,A
                ;
B926 D5                PUSH DE
B927 3E00              LD   A,0
B929 CD1D48            CALL 481DH               ; file close
B92C D1                POP  DE
                ;
B92D 2B                DEC  HL
B92E E5                PUSH HL
B92F EB                EX   DE,HL
B930 E5                PUSH HL
B931 219AB9            LD   HL,TOPADR           ; top address output
B934 CD7BB9            CALL DSPMES
B937 E1                POP  HL
B938 CD4DB9            CALL HEXDSP
B93B CD84B9            CALL CRLF
                ;
B93E 21A7B9            LD   HL,ENDADR           ; end address output
B941 CD7BB9            CALL DSPMES
B944 E1                POP  HL
```

```
B945 CD4DB9             CALL HEXDSP
B948 CD84B9             CALL CRLF
                ;
B94B 18B3               JR   GETADR
                ;
B94D E5         HEXDSP:PUSH HL                  ; print hexadecimal 2 byte data
B94E 7C                 LD   A,H
B94F CD58B9             CALL BYTDSP
B952 7D                 LD   A,L
B953 CD58B9             CALL BYTDSP
B956 E1                 POP  HL
B957 C9                 RET
                ;
B958 E5         BYTDSP:PUSH HL                  ; print hexadecimal 1 byte data
B959 F5                 PUSH AF
B95A E6F0               AND  0F0H               ; mask lower byte
B95C 0F                 RRCA                    ; 1/16
B95D 0F                 RRCA
B95E 0F                 RRCA
B95F 0F                 RRCA
B960 FE0A               CP   10
B962 3802               JR   C,BYTSKP
B964 C607               ADD  A,7                ; A-F
                ;
B966 C630       BYTSKP:ADD  A,'0'               ; convert to ascii
B968 CD0D3E             CALL 3E0DH
B96B F1                 POP  AF
                ;
B96C E60F               AND  0FH                ; mask upper byte
B96E FE0A               CP   10
B970 3802               JR   C,BYTSK2
B972 C607               ADD  A,7
                ;
B974 C630       BYTSK2:ADD  A,'0'
B976 CD0D3E             CALL 3E0DH
B979 E1                 POP  HL
B97A C9                 RET
                ;
B97B 7E         DSPMES:LD   A,(HL)              ; display message subroutine
B97C A7                 AND  A
B97D C8                 RET  Z
B97E CD0D3E             CALL 3E0DH
B981 23                 INC  HL
B982 18F7               JR   DSPMES
                ;
B984 21B4B9     CRLF:   LD   HL,CRLFD           ; line feed
B987 CD7BB9             CALL DSPMES
B98A C9                 RET
                ;
B98B 0D0A4669 FILENM:DEFM 0DH,0AH,'File name ? ',0
B98F 6C65206E
B993 616D6520
B997 3F2000
B99A 546F7020 TOPADR:DEFM 'Top address:',0
B99E 61646472
B9A2 6573733A
B9A6 00
B9A7 456E6420 ENDADR:DEFM 'End address:',0
B9AB 61646472
B9AF 6573733A
B9B3 00
                ;
B9B4 480D0A00 CRLFD: DEFM 'H',0DH,0AH,0
                ;
B9B8                    END


a=&hb900:call a

File name ? machine
Top address:0000H
End address:07FEH
```

```
File name ? sample.bin
Top address:C000H
End address:C0FFH

File name ?
Ok
```

５５５０Ｈ　ライン・アウトプットを行なう．

アドレス	５５５０Ｈ
機　　能	ライン・アウトプットを行なう．
レジスタ	A，F，B，C，D，E，H，L

解　　説　ＥＣ８８Ｈ，８９Ｈ両番地の値が００００Ｈの場合にはＣＲＴ上へ，それ以外の値の場合にはそのアドレスで示されるファイルバッファへ，ＨＬレジスタペアの指すアドレスからそのデータが００Ｈあるいは２２Ｈ（"）になるまで書き込みを行ないます．リターン時にはＨＬレジスタペアは，書き込んだデータの終了アドレスを示しています．

サンプル

```
                ;
                ; --- create sequential file
                ;
                ;       sample of subroutine from 5550H
                ;
                        ORG   0B900H
                ;
B900 2151B9     CRSEQF:LD     HL,FILENM          ; input file number
B903 CD48B9             CALL  DSPMES
B906 CDC85F             CALL  5FC8H              ; line input subroutine
B909 D8                 RET   C                  ; stop key return
                ;
B90A 3622               LD    (HL),'"'
B90C CD8C46             CALL  468CH              ; processing of file descriptor
                ;
B90F 3E01               LD    A,1                ; file#=1
B911 1E02               LD    E,2                ; output mode
B913 CDF647             CALL  47F6H              ; open file
                ;
B916 210000     SLOOP: LD     HL,0
B919 2288EC             LD    (0EC88H),HL        ; clear file buffer address
                ;
B91C 2160B9             LD    HL,STRINP          ; input strings
B91F CD48B9             CALL  DSPMES
B922 CDC85F             CALL  5FC8H              ; line input subroutine
B925 23                 INC   HL
B926 7E                 LD    A,(HL)
B927 B7                 OR    A
B928 2817               JR    Z,EXIT             ; null exit
                ;
B92A 3E01               LD    A,1                ; file#=1
B92C E5                 PUSH  HL
B92D CD3547             CALL  4735H              ; set file buffer address
B930 E1                 POP   HL
                ;
B931 CD5055             CALL  5550H              ; output to file
                ;
B934 3E01               LD    A,1
B936 CD3547             CALL  4735H
                ;
B939 2166B9             LD    HL,CRLF            ; line feed
B93C CD5055             CALL  5550H
                ;
B93F 18D5               JR    SLOOP
```

```
                 ;
B941 3E01        EXIT:  LD   A,1              ; file#=1
B943 CD1D48             CALL 481DH            ; file close
                 ;
B946 18B8               JR   CRSEQF
                 ;
B948 7E          DSPMES:LD   A,(HL)           ; display message subroutine
B949 A7                 AND  A
B94A C8                 RET  Z
B94B CD0D3E             CALL 3E0DH
B94E 23                 INC  HL
B94F 18F7               JR   DSPMES
                 ;
B951 0D0A4669 FILENM:DEFM 0DH,0AH,'File name ? ',0
B955 6C65206E
B959 616D6520




B95D 3F2000
B960 44617461 STRINP:DEFM 'Data:',0
B964 3A00
B966 0D0A00    CRLF:  DEFM 0DH,0AH,0
                 ;
B969                  END
```

```
a=&hb900:call a

File name ? sample.dat
Data:This program is
Data:sample of subroutine from 5550H !
Data:

File name ?
Ok
open "sample.dat" for input as #1
Ok
while not eof(1):line input #1,a$:print a$:wend
This program is
sample of subroutine from 5550H !
Ok
close
Ok
```

5FC8H　ライン・インプットを行う.

アドレス　5FC8H

機　能　ライン・インプットを行う.

レジスタ　A，F，B，C，D，E，H，L

解　説　EC88H，89H番地の値が0000Hの場合にはキーボードから，それ以外の値の場合にはそのアドレスで示されるファイルバッファから，$^{C}{}_{R}$コード（0DH）の入力が認められるまでデータを読み込み，E9B9H番地からのバッファに格納して戻ります．リターン時には，HLレジスタペアにバッファの先頭アドレス－1の値が格納されています．

また，バッファへの入力が中断された場合（キーボードからの入力：STOPキー，ファイルからの入力：END　OF　FILE）には，CYフラグをセットして戻ります．

６Ｆ０６Ｈ　ミニディスクユニットを両面仕様にセットする.

アドレス	６Ｆ０６Ｈ
機　能	ミニディスクユニットを両面仕様にセットする.
レジスタ	Ａ，Ｆ

解　説　接続されているミニディスクユニットがＰＣ－８０３１（片面ドライブ)か，ＰＣ－８０３１－２Ｗ(両面ドライブ)かを調べ，ＰＣ－８０３１－２Ｗであれば，その全てのドライブを両面仕様に設定します．ＰＣ－８０３１－２Ｗには両面モードと片面モードがあり，このサブルーチンでは前者の両面モードに設定しています．

実際に２Ｗモードにセットするルーチンはサブ ROMにあり，４５５１Ｈ番地からのサブルーチン（サブROM内のサブルーチンを呼び出すサブルーチン）を機能番号２８を指定し呼び出すことで，このサブルーチンを実現しています．

サブROM内の処理アドレスは７５Ａ１Ｈ番地です．

サンプル

```
                  ;
                  ; --- set PC-8031-2W double sided
                  ;
                  ;       sample of subroutine from 6F06H
                  ;
                          ORG  0B900H
                  ;
B900 CD066F       SET2WM:CALL 6F06H              ; set 2W mode
                  ;
B903 2113B9               LD   HL,MESS           ; message print
B906 CD0AB9               CALL DSPMES
                  ;
B909 C9                   RET
                  ;
B90A 7E           DSPMES:LD   A,(HL)             ; display message subroutine
B90B A7                   AND  A
B90C C8                   RET  Z
B90D CD0D3E               CALL 3E0DH
B910 23                   INC  HL
B911 18F7                 JR   DSPMES
                  ;
B913 0D0A416C     MESS:   DEFM 0DH,0AH,'All drives of PC-8031-2W'
B917 6C206472
B91B 69766573
B91F 206F6620
B923 50432D38
B927 3033312D
B92B 3257
B92D 20697320             DEFM ' is double sided !!',0
B931 646F7562
B935 6C652073
B939 69646564
B93D 20212100
                  ;
B941                      END
```

```
a=&hb900:call a

All drives of PC-8031-2W is double sided !!
Ok
```

６Ｆ０ＡＨ　ドライブタイプの対応表を作成する．

アドレス	６Ｆ０ＡＨ
機　能	ドライブタイプの対応表をメモリ上に作成する．
レジスタ	Ａ，Ｆ，Ｂ，Ｃ，Ｅ，Ｈ，Ｌ

解　説　接続されている各ドライブの形式を調べ，本体メモリ上に各ドライブ・ナンバに対応したドライブタイプの表を作成します．表の先頭アドレスはＥＦ６４Ｈ番地となっており，ドライブ１から順に対応しています．対応するデータとドライブタイプの関係は以下の様になります．

０　０Ｈ………ＤＭＡ転送方式８インチ
０　１Ｈ………ＤＭＡ転送方式５インチ
０　２Ｈ………インテリジェントタイプ片面
０　３Ｈ………インテリジェントタイプ両面

実際に表を作成するルーチンはサブＲＯＭにあり，４５５１Ｈ番地からのサブルーチン（サブＲＯＭ内の処理を呼び出すサブルーチン）を機能番号２９を指定し呼び出すことで，このサブルーチンを実現しています．

サブＲＯＭ内の処理アドレスは７５４ＦＨ番地です．

サンプル

```
                ;
                ; --- get drive type of each drive
                ;
                ;       sample of subroutine from 6F0AH
                ;
                        ORG   0B900H
                ;
B900 CD0A6F     GETTYP:CALL  6F0AH           ; create type chart
                ;
B903 2164EF             LD    HL,0EF64H       ; top of chart
B906 0E00               LD    C,0             ; top of drive
                ;
B908 E5         GETTPL:PUSH  HL
B909 214CB9             LD    HL,DRVMES
B90C CD43B9             CALL  DSPMES
                ;
B90F 79                 LD    A,C
B910 C631               ADD   A,'1'           ; increment and convert to ascii
B912 CD0D3E             CALL  3E0DH
                ;
B915 2155B9             LD    HL,DRVMS2
B918 CD43B9             CALL  DSPMES
B91B E1                 POP   HL
                ;
B91C E5                 PUSH  HL
B91D 0600               LD    B,0             ; get drive type byte
B91F 09                 ADD   HL,BC
B920 5E                 LD    E,(HL)
                ;
B921 215AB9             LD    HL,MESTBL       ; top of message table
B924 1600               LD    D,0
B926 19                 ADD   HL,DE
```

```
B927 19               ADD  HL,DE
B928 5E               LD   E,(HL)
B929 23               INC  HL
B92A 56               LD   D,(HL)
B92B EB               EX   DE,HL

B92C CD43B9           CALL DSPMES
B92F E1               POP  HL
             ;
B930 0C               INC  C
B931 3A7DEC           LD   A,(0EC7DH)        ; maximum drive number
B934 3D               DEC  A
B935 B9               CP   C
B936 30D0             JR   NC,GETTPL
             ;
B938 3E0D             LD   A,0DH             ; line feed
B93A CD0D3E           CALL 3E0DH
B93D 3E0A             LD   A,0AH
B93F CD0D3E           CALL 3E0DH
             ;
B942 C9               RET
             ;
B943 7E      DSPMES:LD   A,(HL)              ; display message subroutine
B944 A7               AND  A
B945 C8               RET  Z
B946 CD0D3E           CALL 3E0DH
B949 23               INC  HL
B94A 18F7             JR   DSPMES
             ;
B94C 0D0A4472 DRVMES:DEFM 0DH,0AH,'Drive ',0
B950 69766520
B954 00
B955 20697320 DRVMS2:DEFM ' is ',0
B959 00
             ;
B95A 62B9    MESTBL:DEFW DMA8
B95C 70B9           DEFW DMA5
B95E 7EB9           DEFW INTS
B960 99B9           DEFW INTD
             ;
B962 444D4120 DMA8:  DEFM 'DMA 8 inches.',0
B966 3820696E
B96A 63686573
B96E 2E00
B970 444D4120 DMA5:  DEFM 'DMA 5 inches.',0
B974 3520696E
B978 63686573
B97C 2E00
B97E 496E7465 INTS:  DEFM 'Inteligent single surface.',0
B982 6C696765
B986 6E742073
B98A 696E676C
B98E 65207375
B992 72666163
B996 652E00
B999 496E7465 INTD:  DEFM 'Inteligent double surface.',0
B99D 6C696765
B9A1 6E742064
B9A5 6F75626C
B9A9 65207375
B9AD 72666163
B9B1 652E00
             ;
B9B4                  END
```

```
a=&hb900:call a

Drive 1 is DMA 8 inches.
Drive 2 is DMA 8 inches.
Drive 3 is Inteligent double surface.
Drive 4 is Inteligent double surface.
Ok
```

〔サブROMについて〕

PC－8801には拡張ROMを実装することができ，アドレスとしては6000H～7FFFH番地の8Kバイトが割り当てられています．またこの領域には最大8バンクのROMを持つことができ，結局64KバイトのROMを拡張できます．

7FFFH
6000H
0000H
メイン ROM
ROM1　ROM2　----　ROM8

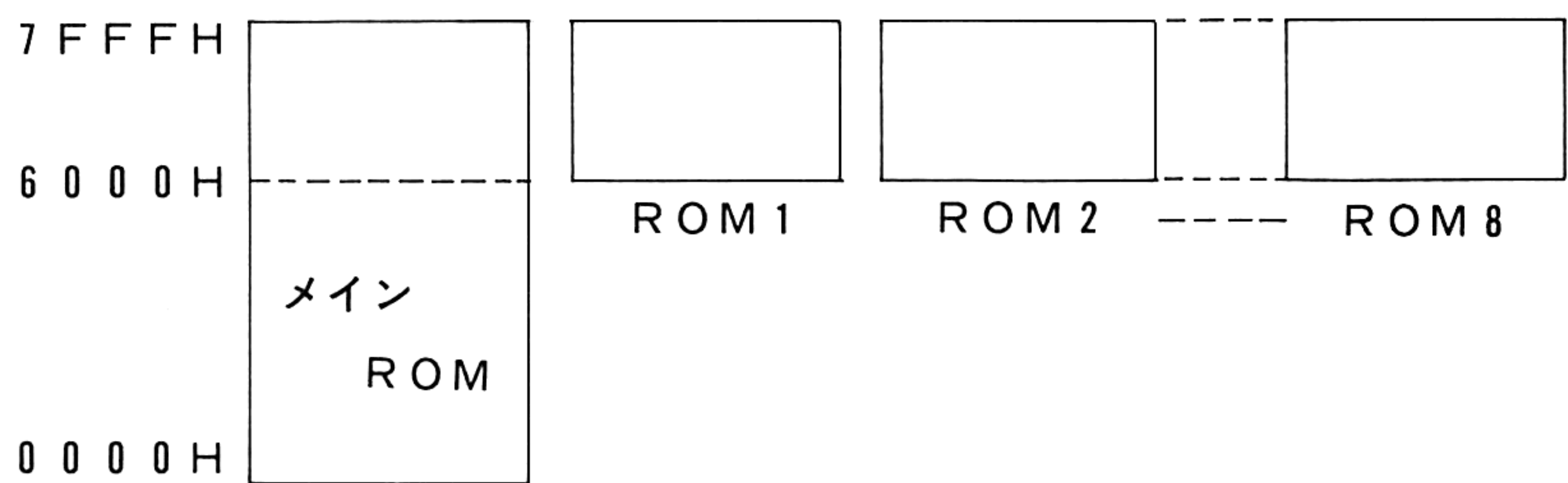



Fig. 3－B－1　ROM配置状況

このうちROM1だけはN_{88}－BASICインタプリタの一部，主にグラフィック処理関係のルーチンが格納されており，実際に拡張できるのは56Kバイトまでです．

各サブROMの選択は，次の様にしてI／Oポート71H番地を使用して行います．

7 6 5 4 3 2 1 0
データ・バイト
ROM8　ROM7　ROM6　ROM5
ROM1　ROM2　ROM3　ROM4

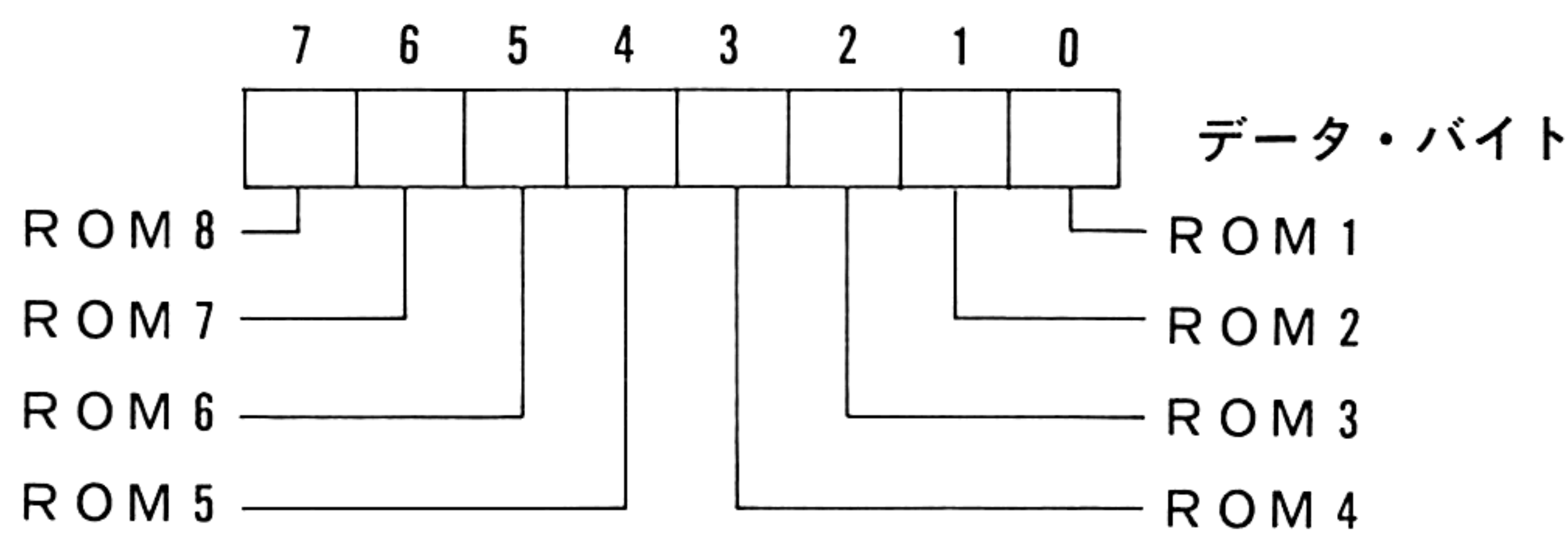



Fig. 3－B－2　ROMセレクト・バイト

このとき，全てのビットは0で意味があり，例えばROM1を選択したい場合には，

```
LD    A，0FEH  ；第0ビットが0
OUT   (71H)，A
```

という具合に行います．他のＲＯＭでも同様です．

また，現在選択されているＲＯＭを知ることもできます．それは入力ポート７１Ｈ番地にＩＮ命令を実行すればよく，各ビットの構成はFig. ３－２と同様です．

〔サブＲＯＭと機能番号について〕

N_{88}－ＢＡＳＩＣでは，サブＲＯＭ内ルーチンの呼び出しを簡略化するために各処理に応じて機能番号を割り当て，それに応じてサブＲＯＭ内で分岐する方法をとっています．また，この処理を行なうサブルーチンがメインＲＯＭ上に用意されており，次の様にしてサブＲＯＭをコールします．

```
CALL  4551H
DEFB  <機能番号>
```

また，各機能ごとにこれらの手続きがメインＲＯＭ上に用意されています．機能番号0から３０まで4バイトづつ６Ｅ９６Ｈ番地から順に並んでいます．６Ｆ０６Ｈと６Ｆ０ＡＨ番地からのサブルーチンは，それぞれ機能番号２８と２９を指定して，サブＲＯＭ内の処理に分岐するものです．

８８Ａ４Ｈ　グラフィック画面を指定データでクリアする．

アドレス　８８Ａ４Ｈ

機　能　グラフィック画面を指定データでクリアする．

レジスタ　Ａ，Ｆ，Ｂ，Ｃ，Ｄ，Ｅ，Ｈ，Ｌ

解　説　グラフィック画面を指定データでクリアするサブルーチンで，ＲＯＬＬ文における画面クリアで使用されているものです．プレーンの指定などのパラメータは以下の様になります．

Ｅ９ＢＢＨ，ＢＣＨ……クリアする領域の先頭アドレス．
Ｆ０３６Ｈ………………プレーン１（青）をクリアするためのデータ．
Ｆ０３７Ｈ………………プレーン２（赤）をクリアするためのデータ．
Ｆ０３８Ｈ………………プレーン３（緑）をクリアするためのデータ．
Ｆ０８９Ｈ………………プレーン・ナンバ（カラーモードのときは５ＣＨ，白黒モードのときはページ・ナンバ＋５ＣＨ）
Ｆ０８ＡＨ　　　　　　スクリーンモード（カラー：３，白黒：１）．
Ｆ０ＡＣＨ，ＡＤＨ　　クリアするバイト数．

このとき，Ｅ９ＢＢＨ，ＢＣＨ両番地で示されるアドレスはＣ０００Ｈ～ＦＦＦＦＨの範囲になければならず，Ｆ０ＡＣＨ，ＡＤＨ両番地で示されるバイト数を加えても，その範囲を越えない様にする必要があります．

サンプル

```
                ;
                ; --- fill graphic RAM with constant
                ;
                ;       sample of subroutine from 88A4H
                ;
                        ORG   0B900H
                ;
B900 3E5C               LD    A,5CH             ; plane 1 (color mode)
B902 3289F0             LD    (0F089H),A
                ;
B905 3E03               LD    A,3               ; graphic mode
B907 328AF0             LD    (0F08AH),A
                ;
B90A 210040             LD    HL,4000H          ; bytes
B90D 22ACF0             LD    (0F0ACH),HL       ; store to work
                ;
B910 2161B9     FLGRAM:LD     HL,FILDAT         ; input data for fill
B913 CD58B9             CALL  DSPMES
B916 CDC85F             CALL  5FC8H             ; line input from keyboard
B919 D8                 RET   C                 ; stop key return
                ;
B91A CD37B9             CALL  GETHEX            ; get plane 1 data
B91D 3236F0             LD    (0F036H),A
                ;
```

```
B920 CD37B9          CALL GETHEX          ; get plane 2 data
B923 3237F0          LD   (0F037H),A
               ;
B926 CD37B9          CALL GETHEX          ; get plane 3 data
B929 3238F0          LD   (0F038H),A
               ;
B92C 2100C0          LD   HL,0C000H       ; top address of GRAM
B92F 22BBE9          LD   (0E9BBH),HL
               ;
B932 CDA488          CALL 88A4H           ; fill GRAM
               ;
B935 18D9            JR   FLGRAM
               ;
B937 0600      GETHEX:LD   B,0            ; get hexadecimal data
B939 23        GETHXL:INC  HL
B93A 7E              LD   A,(HL)          ; take 1 charactor
B93B FE2C            CP   ','
B93D 2817            JR   Z,GETHXE
B93F B7              OR   A
B940 2814            JR   Z,GETHXE
               ;
B942 F5              PUSH AF
B943 78              LD   A,B             ; B*16
B944 87              ADD  A,A
B945 87              ADD  A,A
B946 87              ADD  A,A
B947 87              ADD  A,A
B948 47              LD   B,A
B949 F1              POP  AF
               ;
B94A D630            SUB  '0'             ; convert to binary
B94C FE0A            CP   10
B94E 3802            JR   C,GETHXJ        ; alpha
               ;
B950 D607            SUB  7
B952 80        GETHXJ:ADD  A,B
B953 47              LD   B,A
B954 18E3            JR   GETHXL
               ;
B956 78        GETHXE:LD   A,B
B957 C9              RET
               ;
B958 7E        DSPMES:LD   A,(HL)         ; display message subroutine
B959 A7              AND  A
B95A C8              RET  Z
B95B CD0D3E          CALL 3E0DH
B95E 23              INC  HL
B95F 18F7            JR   DSPMES
               ;
B961 0D0A436F FILDAT:DEFM 0DH,0AH,'Constants(BB,RR,GG) ? ',0
B965 6E737461
B969 6E747328
B96D 42422C52
B971 522C4747
B975 29203F20
B979 00
               ;
B97A                 END
```

```
a=&hb900:call a


Constants(BB,RR,GG) ? 11,22,33

Constants(BB,RR,GG) ? FF,55,AA

Constants(BB,RR,GG) ? FF,FF,FF

Constants(BB,RR,GG) ? 00,00,00

Constants(BB,RR,GG) ?
Ok
```

８８B９H　グラフィック画面のスクロールを行う.

アドレス　８８B９H

機　　能　グラフィック画面のスクロールを行う.

レジスタ　A，F，B，C，D，E，H，L

解　　説　カラーグラフィックモード，白黒モードにおいてグラフィック画面のスクロールを行うサブルーチンで，ＲＯＬＬ文におけるスクロール部分で使用されているものです．プレーンの指定などのパラメータは以下の様になります.

Ｅ９Ｂ９Ｈ，ＢＡＨ……スクロールをする領域の先頭アドレス.

Ｆ０８９Ｈ………………プレーン・ナンバ（カラーモードのときは５ＣＨ，白黒モードのときはページ・ナンバ＋５ＣＨ）.

Ｆ０８ＡＨ………………スクリーンモード（カラー：３，白黒：１）.

Ｆ０ＡＥＨ，ＡＦＨ……スクロールをするバイト数.

このサブルーチンでは，Ｅ９Ｂ９Ｈ，ＢＡＨ両番地で示されるアドレスから，Ｆ０ＡＥＨ，ＡＦＨ両番地で示されるバイト数分の領域を，Ｃ０００Ｈ番地からの領域にスクロールします．またスクロールする前にあったデータはクリアされません.

また，このときＥ９Ｂ９Ｈ，ＢＡＨ両番地で示されるアドレスはＣ０００Ｈ～ＦＦＦＦＨの範囲になければならず，Ｆ０ＡＥ，ＡＦＨ両番地で示されるバイト数を加えても，その範囲を越えない様にする必要があります.

なお，このサブルーチンは高分解能白黒モードでは正しくスクロールしませんので御注意下さい.

サンプル

```
                    ;
                    ; --- scroll graphic screen
                    ;
                    ;       sample of subroutine from 88B9H
                    ;
                            ORG   0B900H
                    ;
B900 3E5C                   LD    A,5CH              ; defolt plane number
B902 3289F0                 LD    (0F089H),A         ; set data to work
                    ;
B905 3E03                   LD    A,3                ; color mode
B907 328AF0                 LD    (0F08AH),A
                    ;
B90A 3A87F0                 LD    A,(0F087H)         ; check screen mode
B90D B7                     OR    A
B90E 280F                   JR    Z,ROLGRH           ; color mode
                    ;
B910 FE02                   CP    2
```

```
B912 284E               JR    Z,NOTROL            ; figh-reso mode
                ;
B914 3A8BF0             LD    A,(0F08BH)          ; active page port
B917 3289F0             LD    (0F089H),A
B91A 3E01               LD    A,1                 ; B/W mode
B91C 328AF0             LD    (0F08AH),A
                ;
B91F 2172B9     ROLGRH:LD     HL,ROLBYT           ; input scroll lines
B922 CD69B9             CALL  DSPMES
B925 CDC85F             CALL  5FC8H               ; line input from keyboard
B928 D8                 RET   C                   ; stop key return
                ;
B929 23                 INC   HL
B92A CDBC26             CALL  26BCH               ; convert to binary
B92D CDA021             CALL  21A0H               ; convert to integer
B930 7D                 LD    A,L
B931 FEC8               CP    200
B933 30EA               JR    NC,ROLGRH
                ;
B935 B7                 OR    A
B936 28E7               JR    Z,ROLGRH
                ;
B938 7C                 LD    A,H
B939 B7                 OR    A
B93A 20E3               JR    NZ,ROLGRH
                ;
B93C EB                 EX    DE,HL
B93D 215000             LD    HL,80               ; bytes a line
B940 CD5A23             CALL  235AH               ; HL*DE
                ;
B943 E5                 PUSH  HL
B944 1100C0             LD    DE,0C000H           ; top of GRAM
B947 19                 ADD   HL,DE
B948 22B9E9             LD    (0E9B9H),HL         ; scroll address
B94B E1                 POP   HL
                ;
B94C EB                 EX    DE,HL
B94D 21803E             LD    HL,16000            ; bytes all rams
B950 B7                 OR    A                   ; clear carry
B951 ED52               SBC   HL,DE
B953 22AEF0             LD    (0F0AEH),HL
                ;
B956 1100C0             LD    DE,0C000H
B959 19                 ADD   HL,DE
B95A 22BBE9             LD    (0E9BBH),HL
                ;
B95D CDB988             CALL  88B9H               ; scroll
                ;
B960 18BD               JR    ROLGRH
                ;
B962 218CB9     NOTROL:LD     HL,NOTMES           ; disenable message
B965 CD69B9             CALL  DSPMES
B968 C9                 RET
                ;
B969 7E         DSPMES:LD     A,(HL)              ; display message subroutine
B96A A7                 AND   A
B96B C8                 RET   Z
B96C CD0D3E             CALL  3E0DH
B96F 23                 INC   HL
B970 18F7               JR    DSPMES
                ;
B972 0D0A5363   ROLBYT:DEFM 0DH,0AH,'Scroll lines (1-199) ? ',0
B976 726F6C6C
B97A 206C696E
B97E 65732028
B982 312D3139
B986 3929203F
B98A 2000
                ;
B98C 0D0A4361   NOTMES:DEFM 0DH,0AH,'Can',27H,'t scroll !!',7,0
B990 6E277420
B994 7363726F
B998 6C6C2021
B99C 210700
```

```
                    ;
B99F                           END


k=0:for i=0 to 620 step 18:put@(i,184),kanji(&h3021+k):k=k+1:next
Ok
a=&hb900:call a

Scroll lines (1-199) ? 16

Scroll lines (1-199) ? 16

Scroll lines (1-199) ? 16

Scroll lines (1-199) ? 16

Scroll lines (1-199) ?
Ok
copy 5
亜唖娃阿哀愛挨姶逢葵茜穐悪握渥旭葦芦鯵梓圧斡扱宛姐虻飴絢綾鮎或粟袷安庵
亜唖娃阿哀愛挨姶逢葵茜穐悪握渥旭葦芦鯵梓圧斡扱宛姐虻飴絢綾鮎或粟袷安庵
亜唖娃阿哀愛挨姶逢葵茜穐悪握渥旭葦芦鯵梓圧斡扱宛姐虻飴絢綾鮎或粟袷安庵
亜唖娃阿哀愛挨姶逢葵茜穐悪握渥旭葦芦鯵梓圧斡扱宛姐虻飴絢綾鮎或粟袷安庵
亜唖娃阿哀愛挨姶逢葵茜穐悪握渥旭葦芦鯵梓圧斡扱宛姐虻飴絢綾鮎或粟袷安庵
```

８９Ｂ３Ｈ　単精度型実数の逆正接値を求める.

アドレス	８９Ｂ３Ｈ
機　　能	単精度型実数の逆正接値を求める.
レジスタ	Ａ，Ｆ，Ｂ，Ｃ，Ｄ，Ｅ，Ｈ，Ｌ

解　　説　フローティングポイント・アキュムレータ（ＦＡＣ）に格納された単精度型実数の値をラジアンとした場合の逆正接値（アークタンジェント）を求め，結果を同じくフローティングポイント・アキュムレータ（ＦＡＣ）に返します.

フローティングポイント・アキュムレータ（ＦＡＣ）に与えられた値が倍精度型実数であった場合，倍精度の値を返します.

サンプル

```
                ;
                ; --- arc tangent function (single precision) on FAC
                ;
                ;       sample of subroutine from 89B3H
                ;
                        ORG   0B900H
                ;
EC3D            FACPNT:EQU   0EC3DH           ; floating point accumlator
                ;
B900 214BB9     ATNSNG:LD    HL,PROMPT        ; input number
B903 CD42B9             CALL  DSPMES
B906 CDC35F             CALL  5FC3H            ; line input
B909 D8                 RET   C                ; if stop key return
                ;
B90A 23                 INC   HL
B90B E5                 PUSH  HL
                ;
B90C 110000             LD    DE,0             ; clear FAC
B90F ED533DEC           LD    (FACPNT),DE
B913 ED533FEC           LD    (FACPNT+2),DE
B917 ED5341EC           LD    (FACPNT+4),DE
B91B ED5343EC           LD    (FACPNT+6),DE
                ;
B91F CDC826             CALL  26C8H            ; charactor code to binary code
B922 CD1422             CALL  2214H            ; convert FAC to single precision
B925 CDB389             CALL  89B3H            ; calc. arc tangent (single prec.)
B928 CDD028             CALL  28D0H            ; binary code on FAC to charactor
                ;
B92B EB                 EX    DE,HL
B92C 2168B9             LD    HL,ANSWE1
B92F CD42B9             CALL  DSPMES
                ;
B932 E1                 POP   HL
B933 CD42B9             CALL  DSPMES           ; display number
                ;
B936 2175B9             LD    HL,ANSWE2
B939 CD42B9             CALL  DSPMES
                ;
B93C EB                 EX    DE,HL
B93D CD42B9             CALL  DSPMES
                ;
B940 18BE               JR    ATNSNG
```

```
                ;
                ; subroutine display message
                ;
B942 7E         DSPMES:LD   A,(HL)
B943 A7                AND  A
B944 C8                RET  Z
B945 CD0D3E            CALL 3E0DH            ; one charactor output to CRT
B948 23                INC  HL
B949 18F7              JR   DSPMES
                ;
                ; message area
                ;
B94B 0D0A5369 PROMPT:DEFM 0DH,0AH,'Single precision number ? ',0
B94F 6E676C65
B953 20707265
B957 63697369
B95B 6F6E206E
B95F 756D6265




B963 72203F20
B967 00
B968 41726320 ANSWE1:DEFM 'Arc tangent(',0
B96C 74616E67
B970 656E7428
B974 00
B975 20293D20 ANSWE2:DEFM ' )= ',0
B979 00
                ;
B97A                   END
```

```
a=&hb900:call a

Single precision number ?  0
Arc tangent(0 )=  0
Single precision number ?  .5
Arc tangent(.5 )=  .463648
Single precision number ?  10
Arc tangent(10 )=  1.47113
Single precision number ?  5000
Arc tangent(5000 )=  1.5706
Single precision number ?
Ok
```

９４８ＢＨ　倍精度型実数の逆正接値を求める.

アドレス　９４８ＢＨ

機　　能　倍精度型実数の逆正接値を求める.

レジスタ　A, F, B, C, D, E, H, L

解　　説　フローティングポイント・アキュムレータ（ＦＡＣ）に格納された倍精度型実数の値をラジアンとした場合の逆正接値（アークタンジェント）を求め，結果を同じくフローティングポイント・アキュムレータ（ＦＡＣ）に返します.

フローティングポイント・アキュムレータに与えられた値が倍精度型実数でなかった場合，結果が正しく返されませんので御注意下さい.

サンプル

```
                ;
                ; --- arc tangent function (double precision) on FAC
                ;
                ;       sample of subroutine from 948BH
                ;
                        ORG   0B900H
                ;
EC3D            FACPNT:EQU   0EC3DH                 ; floating point accumlator
                ;
B900 214BB9     ATNDBL:LD    HL,PROMPT              ; input number
B903 CD42B9             CALL  DSPMES
B906 CDC85F             CALL  5FC8H                 ; line input
B909 D8                 RET   C                     ; if stop key return
                ;
B90A 23                 INC   HL
B90B E5                 PUSH  HL
                ;
B90C 110000             LD    DE,0                  ; clear FAC
B90F ED533DEC           LD    (FACPNT),DE
B913 ED533FEC           LD    (FACPNT+2),DE
B917 ED5341EC           LD    (FACPNT+4),DE
B91B ED5343EC           LD    (FACPNT+6),DE
                ;
B91F CDC826             CALL  26C8H                 ; charactor code to binary code
B922 CD3E22             CALL  223EH                 ; convert to double precision
B925 CD8B94             CALL  948BH                 ; calc. arc tangent (double prec.)
B928 CDD028             CALL  28D0H                 ; binary code on FAC to character
                ;
B92B EB                 EX    DE,HL
B92C 2168B9             LD    HL,ANSWE1
B92F CD42B9             CALL  DSPMES
                ;
B932 E1                 POP   HL
B933 CD42B9             CALL  DSPMES                ; display number
                ;
B936 2176B9             LD    HL,ANSWE2
B939 CD42B9             CALL  DSPMES
                ;
B93C EB                 EX    DE,HL
B93D CD42B9             CALL  DSPMES
                ;
B940 18BE               JR    ATNDBL
```

```
                    ;
                    ; subroutine display message
                    ;
B942 7E        DSPMES:LD    A,(HL)
B943 A7               AND   A
B944 C8               RET   Z              ; if end code then return
B945 CD0D3E           CALL  3E0DH          ; one character output to CRT
B948 23               INC   HL
B949 18F7             JR    DSPMES
               ;
               ; message area
               ;
B94B 0D0A446F PROMPT:DEFM 0DH,0AH,'Double precision number ? ',0
B94F 75626C65
B953 20707265
B957 63697369
B95B 6F6E206E
B95F 756D6265
B963 72203F20
B967 00
B968 41726320 ANSWE1:DEFM 'Arc tangent( ',0
B96C 74616E67
B970 656E7428
B974 2000
B976 20293D20 ANSWE2:DEFM ' )= ',0
B97A 00
               ;
B97B                  END
```

```
a=&hb900:call a

Double precision number ? 0
Arc tangent( 0 )=  0
Double precision number ? .5
Arc tangent( .5 )=  .4636476090008062
Double precision number ? 10
Arc tangent( 10 )=  1.471127674303735
Double precision number ? 5000
Arc tangent( 5000 )=  1.570596326797563
Double precision number ?
Ok
```

９４Ｅ７Ｈ　倍精度型実数の余弦値を求める．

アドレス	９４Ｅ７Ｈ
機　能	倍精度型実数の余弦値を求める．
レジスタ	Ａ，Ｆ，Ｂ，Ｃ，Ｄ，Ｅ，Ｈ，Ｌ

解　説　フローティングポイント・アキュムレータ（ＦＡＣ）に格納された倍精度型実数の値をラジアンとした場合の余弦値（コサイン）を求め，結果を同じくフローティングポイント・アキュムレータ（ＦＡＣ）に返します．

フローティングポイント・アキュムレータに与えられた値が倍精度型実数でなかった場合，結果が正しく返されませんので御注意下さい．

サンプル

```
                ;
                ; --- cosine function (double precision) on FAC
                ;
                ;       sample of subroutine from 94E7H
                ;
                        ORG   0B900H
                ;
EC3D            FACPNT:EQU   0EC3DH             ; floating point accumlator
                ;
B900 214BB9     COSDBL:LD    HL,PROMPT          ; prompt display
B903 CD42B9            CALL  DSPMES
B906 CDC85F            CALL  5FC8H              ; line input
B909 D8                RET   C                  ; if stop key return
                ;
B90A 23                INC   HL
B90B E5                PUSH  HL
                ;
B90C 110000            LD    DE,0               ; clear FAC
B90F ED533DEC          LD    (FACPNT),DE
B913 ED533FEC          LD    (FACPNT+2),DE
B917 ED5341EC          LD    (FACPNT+4),DE
B91B ED5343EC          LD    (FACPNT+6),DE
                ;
B91F CDC826            CALL  26C8H              ; charactor code to binary code
B922 CD3E22            CALL  223EH              ; convert to double precision
B925 CDE794            CALL  94E7H              ; calculate cosine (double prec.)
B928 CDD028            CALL  28D0H              ; binary code on FAC to character
                ;
B92B EB                EX    DE,HL
B92C 2168B9            LD    HL,ANSWE1
B92F CD42B9            CALL  DSPMES
                ;
B932 E1                POP   HL
B933 CD42B9            CALL  DSPMES             ; display number
                ;
B936 2171B9            LD    HL,ANSWE2
B939 CD42B9            CALL  DSPMES
                ;
B93C EB                EX    DE,HL
B93D CD42B9            CALL  DSPMES
                ;
B940 18BE              JR    COSDBL
                ;
                ; subroutine display message
                ;
B942 7E         DSPMES:LD    A,(HL)
```

```
B943 A7                AND  A
B944 C8                RET  Z                      ; if end code then return
B94B 0D0A446F PROMPT:DEFM 0DH,0AH,'Double precision number ? ',0
B94F 75626C65
B953 20707265
B957 63697369
B95B 6F6E206E
B95F 756D6265
B963 72203F20
B967 00
B968 436F7369 ANSWE1:DEFM 'Cosine( ',0
B96C 6E652820
B970 00
B971 20293D20 ANSWE2:DEFM ' )= ',0
B975 00
              ;
B976                   END
```

```
a=&hb900:call a

Double precision number ? 0
Cosine( 0 )=  1
Double precision number ? 1.57
Cosine( 1.57 )=  7.963267107334255D-04
Double precision number ? 3.14
Cosine( 3.14 )= -.9999987317275395
Double precision number ?
Ok
```

９４Ｆ０Ｈ　倍精度型実数の指数関数の値を求める.

アドレス　９４Ｆ０Ｈ

機　能　倍精度型実数の自然対数の底ｅに対する指数関数の値を求める.

レジスタ　Ａ，Ｆ，Ｂ，Ｃ，Ｄ，Ｅ，Ｈ，Ｌ

解　説　フローティングポイント・アキュムレータ（ＦＡＣ）に格納された倍精度型実数の値の自然対数の底ｅに対する指数関数の値を求め，結果を同じくフローティングポイント・アキュムレータ（ＦＡＣ）に返します.

フローティングポイント・アキュムレータに与えられた値が倍精度型実数でなかった場合，結果が正しく返されませんので御注意下さい.

サンプル

```
                    ;
                    ; --- exponent function (double precision) on FAC
                    ;
                    ;       sample of subroutine from 94F0H
                    ;
                            ORG  0B900H
                    ;
EC3D                FACPNT:EQU  0EC3DH              ; floating point accumlator
                    ;
B900 214BB9         EXPDBL:LD   HL,PROMPT           ; prompt display
B903 CD42B9                 CALL DSPMES
B906 CDC85F                 CALL 5FC8H              ; line input
B909 D8                     RET  C                  ; if stop key return
                    ;
B90A 23                     INC  HL
B90B E5                     PUSH HL
                    ;
B90C 110000                 LD   DE,0               ; clear FAC
B90F ED533DEC               LD   (FACPNT),DE
B913 ED533FEC               LD   (FACPNT+2),DE
B917 ED5341EC               LD   (FACPNT+4),DE
B91B ED5343EC               LD   (FACPNT+6),DE
                    ;
B91F CDC826                 CALL 26C8H              ; charactor code to binary code
B922 CD3E22                 CALL 223EH              ; convert to double precision
B925 CDF094                 CALL 94F0H              ; calc. exponent (double prec.)
B928 CDD028                 CALL 28D0H              ; binary code on FAC to character
                    ;
B92B EB                     EX   DE,HL
B92C 2168B9                 LD   HL,ANSWE1
B92F CD42B9                 CALL DSPMES
                    ;
B932 E1                     POP  HL
B933 CD42B9                 CALL DSPMES             ; display number
                    ;
B936 2173B9                 LD   HL,ANSWE2
B939 CD42B9                 CALL DSPMES
                    ;
B93C EB                     EX   DE,HL
B93D CD42B9                 CALL DSPMES
                    ;
B940 18BE                   JR   EXPDBL
                    ;
                    ; subroutine display message
```

```
                  ;
B942 7E          DSPMES:LD   A,(HL)
B943 A7                 AND  A
B944 C8                 RET  Z                      ; if end code then return
B945 CD0D3E             CALL 3E0DH                  ; one character output to CRT
B948 23                 INC  HL
B949 18F7               JR   DSPMES
                  ;
                  ; message area
                  ;
B94B 0D0A446F PROMPT:DEFM 0DH,0AH,'Double precision number ? ',0
B94F 75626C65
B953 20707265
B957 63697369
B95B 6F6E206E
B95F 756D6265
B963 72203F20
B967 00
B968 4578706F ANSWE1:DEFM 'Expotant( ',0
B96C 74616E74
B970 282000
B973 20293D20 ANSWE2:DEFM ' )= ',0
B977 00
                  ;
B978                    END
```

```
a=&hb900:call a

Double precision number ? 0
Expotant( 0 )=  1
Double precision number ? 3
Expotant( 3 )=  20.08553692318767
Double precision number ? -50
Expotant( -50 )=  1.928749847963918D-22
Double precision number ? 100
OV
Expotant( 100 )=  1.701411834604693D+38
Double precision number ?
Ok
```

９５６ＦＨ　倍精度型実数の自然対数の値を求める．

アドレス　９５６ＦＨ

機　能　倍精度型実数の自然対数の値を求める．

レジスタ　Ａ，Ｆ，Ｂ，Ｃ，Ｄ，Ｅ，Ｈ，Ｌ

解　説　フローティングポイント・アキュムレータ（ＦＡＣ）に格納された倍精度型実数の値の自然対数の値を求め，結果を同じくフローティングポイント・アキュムレータ（ＦＡＣ）に返します．フローティングポイント・アキュムレータに与えた値が０，または負であった場合には Illegal function call エラーが発生します．

フローティングポイント・アキュムレータに与えられた値が倍精度型実数でなかった場合，結果が正しく返されませんので御注意下さい．

サンプル

```
                ;
                ; --- logarithm funtion (double precision) on FAC
                ;
                ;       sample of subroutine from 956FH
                ;
                        ORG    0B900H
                ;
EC3D            FACPNT:EQU     0EC3DH          ; floating point accumlator
                ;
B900 2158B9     LOGDBL:LD      HL,PROMPT       ; prompt display
B903 CD4FB9            CALL    DSPMES
B906 CDC85F            CALL    5FC8H           ; line input
B909 D8                RET     C               ; if stop key return
                ;
B90A 23                INC     HL
B90B E5                PUSH    HL
                ;
B90C 110000            LD      DE,0            ; clear FAC
B90F ED533DEC          LD      (FACPNT),DE
B913 ED533FEC          LD      (FACPNT+2),DE
B917 ED5341EC          LD      (FACPNT+4),DE
B91B ED5343EC          LD      (FACPNT+6),DE
                ;
B91F CDC826            CALL    26C8H           ; charactor code to binary code
B922 CD3E22            CALL    223EH           ; convert to double precision
                ;
B925 EF         CHCPLS:RST     28H             ; check sign of accumlator
B926 FE01              CP      1
B928 2808              JR      Z,CHKOK
B92A 3E07              LD      A,7             ; beep
B92C CD0D3E            CALL    3E0DH
B92F E1                POP     HL
B930 18CE              JR      LOGDBL
                ;
B932 CD6F95     CHKOK: CALL    956FH           ; calc. logarithm (double prec.)
B935 CDD028            CALL    28D0H           ; binary code on FAC to character
                ;
B938 EB                EX      DE,HL
B939 2186B9            LD      HL,ANSWE1
B93C CD4FB9            CALL    DSPMES
```

```
                ;
B93F E1                 POP  HL
B940 CD4FB9             CALL DSPMES             ; display number
                ;
B943 2199B9             LD   HL,ANSWE2
B946 CD4FB9             CALL DSPMES
                ;
B949 EB                 EX   DE,HL
B94A CD4FB9             CALL DSPMES
                ;
B94D 18B1               JR   LOGDBL
                ;
                ; subroutine display message
                ;
B94F 7E         DSPMES:LD   A,(HL)
B950 A7                 AND  A
B951 C8                 RET  Z                  ; if end code then return
B952 CD0D3E             CALL 3E0DH              ; onr character output to CRT
B955 23                 INC  HL
B956 18F7               JR   DSPMES
                ;
                ; message area
                ;
B958 0D0A446F PROMPT:DEFM 0DH,0AH,'Double precision number '
B95C 75626C65
B960 20707265
B964 63697369
B968 6F6E206E
B96C 756D6265
B970 7220
B972 28677265           DEFM '(greater than 0) ? ',0
B976 61746572
B97A 20746861
B97E 6E203029
B982 203F2000
B986 4C6F6761 ANSWE1:DEFM 'Logalithm nature( ',0
B98A 6C697468
B98E 6D206E61
B992 74757265
B996 282000
B999 20293D20 ANSWE2:DEFM ' )= ',0
B99D 00
                ;
B99E                    END
```

```
a=&hb900:call a

Double precision number (greater than 0) ? 1
Logalithm nature( 1 )= -6.938893903907229D-17
Double precision number (greater than 0) ? 2.82
Logalithm nature( 2.82 )=  1.036736884950022
Double precision number (greater than 0) ? 0

Double precision number (greater than 0) ? -1

Double precision number (greater than 0) ? 100
Logalithm nature( 100 )=  4.605170185988092
Double precision number (greater than 0) ?
Ok
```

９６１５Ｈ　倍精度型実数の正弦値を求める．

アドレス	９６１５Ｈ
機　　能	倍精度型実数の正弦値を求める．
レジスタ	Ａ，Ｆ，Ｂ，Ｃ，Ｄ，Ｅ，Ｈ，Ｌ

解　　説　フローティングポイント・アキュムレータ（ＦＡＣ）に格納された倍精度型実数の値をラジアンとした場合の正弦値（サイン）を求め，結果を同じくフローティングポイント・アキュムレータ（ＦＡＣ）に返します．

フローティングポイント・アキュムレータに与えられた値が倍精度型実数でなかった場合，結果が正しく返されませんので御注意下さい．

サンプル

```
               ;
               ; --- sine function (double precision) on FAC
               ;
                       ORG   0B900H
               ;
EC3D           FACPNT:EQU    0EC3DH              ; floating point accumlator
               ;
B900 214BB9    SINDBL:LD     HL,PROMPT           ; input number
B903 CD42B9            CALL  DSPMES
B906 CDC85F            CALL  5FC8H               ; line input
B909 D8                RET   C                   ; if stop key return
               ;
B90A 23                INC   HL
B90B E5                PUSH  HL
               ;
B90C 110000            LD    DE,0                ; clear FAC
B90F ED533DEC          LD    (FACPNT),DE
B913 ED533FEC          LD    (FACPNT+2),DE
B917 ED5341EC          LD    (FACPNT+4),DE
B91B ED5343EC          LD    (FACPNT+6),DE
               ;
B91F CDC826            CALL  26C8H               ; charactor code to binary code
B922 CD3E22            CALL  223EH               ; convert to double precision
B925 CD1596            CALL  9615H               ; calculate sine(double precision)
B928 CDD028            CALL  28D0H               ; binary code on FAC to character
               ;
B92B EB                EX    DE,HL
B92C 2168B9            LD    HL,ANSWE1
B92F CD42B9            CALL  DSPMES
               ;
B932 E1                POP   HL
B933 CD42B9            CALL  DSPMES              ; display number
               ;
B936 216FB9            LD    HL,ANSWE2
B939 CD42B9            CALL  DSPMES
               ;
B93C EB                EX    DE,HL
B93D CD42B9            CALL  DSPMES
               ;
B940 18BE              JR    SINDBL
               ;
               ; subroutine display message
               ;
B942 7E        DSPMES:LD     A,(HL)
```

```
B943 A7                AND   A
B944 C8                RET   Z              ; if end code then return
B945 CD0D3E            CALL  3E0DH          ; one character output to CRT
B948 23                INC   HL
B949 18F7              JR    DSPMES
                ;
                ; message area
                ;
B94B 0D0A446F PROMPT:DEFM 0DH,0AH,'Double precision number ? ',0
B94F 75626C65
B953 20707265
B957 63697369
B95B 6F6E206E
B95F 756D6265
B963 72203F20
B967 00
B968 53696E65 ANSWE1:DEFM 'Sine( ',0
B96C 282000
B96F 20293D20 ANSWE2:DEFM ' )= ',0
B973 00
                ;
B974                   END
```

```
a=&hb900:call a

Double precision number ? 0
Sine( 0 )=  0
Double precision number ? 1.57
Sine( 1.57 )=  .9999996829318346
Double precision number ? 3.14
Sine( 3.14 )=  1.592652916487153D-03
Double precision number ?
Ok
```

９６５０Ｈ　倍精度型実数の平方根を求める．

アドレス　９６５０Ｈ

機　　能　倍精度型実数の平方根を求める．

レジスタ　A，F，B，C，D，E，H，L

解　　説　フローティングポイント・アキュムレータ（ＦＡＣ）に格納された倍精度型実数の値の平方根（スクエアルート）を求め，結果を同じくフローティングポイント・アキュムレータ（ＦＡＣ）に返します．フローティングポイント・アキュムレータに与えられた値が負であった場合には，Illegal function call エラーが発生します．

フローティングポイント・アキュムレータに与えられた値が倍精度型実数でなかった場合，結果が正しく返されませんので御注意下さい．

サンプル

```
                ;
                ; --- square root function (double precision) on FAC
                ;
                ;       sample of subroutine from 9650H
                ;
                        ORG   0B900H
                ;
EC3D            FACPNT:EQU   0EC3DH              ; floating point accumlator
                ;
B900 2157B9     SQRDBL:LD    HL,PROMPT           ; prompt display
B903 CD4EB9            CALL  DSPMES
B906 CDC85F            CALL  5FC8H               ; line input
B909 D8                RET   C                   ; if stop key return
                ;
B90A 23                INC   HL
B90B E5                PUSH  HL
                ;
B90C 110000            LD    DE,0                ; clear FAC
B90F ED533DEC          LD    (FACPNT),DE
B913 ED533FEC          LD    (FACPNT+2),DE
B917 ED5341EC          LD    (FACPNT+4),DE
B91B ED5343EC          LD    (FACPNT+6),DE
                ;
B91F CDC826            CALL  26C8H               ; charactor code to binary code
B922 CD3E22            CALL  223EH               ; convert to double precision
                ;
B925 EF                RST   28H                 ; check sign of accumlator
B926 3C                INC   A
B927 2008              JR    NZ,CHKOK
B929 3E07              LD    A,7                 ; beep
B92B CD0D3E            CALL  3E0DH
B92E E1                POP   HL
B92F 18CF              JR    SQRDBL
                ;
B931 CD5096     CHKOK: CALL  9650H               ; calc. square root (double prec.)
B934 CDD028            CALL  28D0H               ; binary code on FAC to character
                ;
B937 EB                EX    DE,HL
B938 2180B9            LD    HL,ANSWE1
B93B CD4EB9            CALL  DSPMES
```

```
                ;
B93E E1                 POP   HL
B93F CD4EB9             CALL  DSPMES            ; display number
                ;
B942 218EB9             LD    HL,ANSWE2
B945 CD4EB9             CALL  DSPMES
                ;
B948 EB                 EX    DE,HL
B949 CD4EB9             CALL  DSPMES
                ;
B94C 18B2               JR    SQRDBL
                ;
                ; subroutine display message
                ;
B94E 7E         DSPMES:LD    A,(HL)
B94F A7                 AND   A
B950 C8                 RET   Z                 ; if end code then return
B951 CD0D3E             CALL  3E0DH             ; one character output to CRT
B954 23                 INC   HL
B955 18F7               JR    DSPMES
                ;
                ; message area
                ;
B957 0D0A446F   PROMPT:DEFM 0DH,0AH,'Double precision number '
B95B 75626C65
B95F 20707265
B963 63697369
B967 6F6E206E
B96B 756D6265
B96F 7220
B971 286E6F74           DEFM '(not minus) ? ',0
B975 206D696E
B979 75732920
B97D 3F2000
B980 53717561   ANSWE1:DEFM 'Square root( ',0
B984 72652072
B988 6F6F7428
B98C 2000
B98E 20293D20   ANSWE2:DEFM ' )= ',0
B992 00
                ;
B993                    END
```

```
a=&hb900:call a

Double precision number (not minus) ? 2
Square root( 2 )=  1.414213562373095
Double precision number (not minus) ? 3
Square root( 3 )=  1.732050807568877
Double precision number (not minus) ? 4
Square root( 4 )=  2
Double precision number (not minus) ? -2

Double precision number (not minus) ? 5
Square root( 5 )=  2.23606797749979
Double precision number (not minus) ?
Ok
```

９６９１Ｈ　倍精度型実数の正接値を求める．

アドレス	９６９１Ｈ
機　　能	倍精度型実数の正接値を求める．
レジスタ	A, F, B, C, D, E, H, L

解　　説　フローティングポイント・アキュムレータ（ＦＡＣ）に格納された倍精度型実数の値をラジアンとした場合の正接値（タンジェント）を求め，結果を同じくフローティングポイント・アキュムレータ（ＦＡＣ）に返します．

フローティングポイント・アキュムレータに与えられた値が倍精度型実数でなかった場合，結正しく返されませんので御注意下さい．

サンプル

```
                ;
                ; --- tangent function (double precision) on FAC
                ;
                ;       sample of subroutine from 9691H
                ;
                        ORG   0B900H
                ;
EC3D            FACPNT:EQU   0EC3DH               ; floating point accumlator
                ;
B900 214BB9     TANDBL:LD    HL,PROMPT            ; prompt display
B903 CD42B9            CALL  DSPMES
B906 CDC85F            CALL  5FC8H                ; line input
B909 D8                RET   C                    ; if stop key return
                ;
B90A 23                INC   HL
B90B E5                PUSH  HL
                ;
B90C 110000            LD    DE,0                 ; clear FAC
B90F ED533DEC          LD    (FACPNT),DE
B913 ED533FEC          LD    (FACPNT+2),DE
B917 ED5341EC          LD    (FACPNT+4),DE
B91B ED5343EC          LD    (FACPNT+6),DE
                ;
B91F CDC826            CALL  26C8H                ; charactor code to binary code
B922 CD3E22            CALL  223EH                ; convert to double precision
B925 CD9196            CALL  9691H                ; calculate tangent (double prec.)
B928 CDD028            CALL  28D0H                ; binary code on FAC to character
                ;
B92B EB                EX    DE,HL
B92C 2168B9            LD    HL,ANSWE1
B92F CD42B9            CALL  DSPMES
                ;
B932 E1                POP   HL
B933 CD42B9            CALL  DSPMES               ; display number
                ;
B936 2172B9            LD    HL,ANSWE2
B939 CD42B9            CALL  DSPMES
                ;
B93C EB                EX    DE,HL
B93D CD42B9            CALL  DSPMES
                ;
B940 18BE              JR    TANDBL
                ;
                ; subroutine display message
```

```
                  ;
B942 7E          DSPMES:LD    A,(HL)
B943 A7                 AND   A
B944 C8                 RET   Z                ; if end code then return
B945 CD0D3E             CALL  3E0DH            ; one character output to CRT
B948 23                 INC   HL
B949 18F7               JR    DSPMES
                  ;
                  ; message area
                  ;
B94B 0D0A446F PROMPT:DEFM 0DH,0AH,'Double precision number ? ',0
B94F 75626C65
B953 20707265
B957 63697369
B95B 6F6E206E
B95F 756D6265
B963 72203F20
B967 00
B968 54616E67 ANSWE1:DEFM 'Tangent( ',0
B96C 656E7428
B970 2000
B972 20293D20 ANSWE2:DEFM ' )= ',0
B976 00
                  ;
B977                    END
```

```
a=&hb900:call a

Double precision number ? 0
Tangent( 0 )=  0
Double precision number ? 1
Tangent( 1 )=  1.557407724654902
Double precision number ? 1.57
Tangent( 1.57 )=  1255.765591500534
Double precision number ? 3.14
Tangent( 3.14 )= -1.592654936407547D-03
Double precision number ?
Ok
```

〔フローティングポイント・アキュムレータ（ＦＡＣ）の構成〕

Ｚ－８０（μＰＤ７８０）を初めとする８０系ＣＰＵの多くは，浮動小数点演算の機能が用意されていません．そこで，N_{88}－ＢＡＳＩＣでは浮動小数点演算を可能とするために，メモリ上に８バイトからなる仮想レジスタを用意し，これを介してソフトウェア処理による演算処理を行っています．これがフローティングポイント・アキュムレータ（ＦＡＣ）と呼ばれているもので，N_{88}－ＢＡＳＩＣではＥＣ３ＤＨ～ＥＣ４４Ｈ番地が割り当てられています．

また，ＥＡＢＤＨ番地にはＦＡＣ内のデータの型を示す数値が格納されており，その値によりＦＡＣは異なる形式の数値を示していることになります．以下にＥＡＢＤＨ番地の値によるＦＡＣの区分を示します．

ＥＡＢＤＨ＝０２Ｈ（整数型）

ＥＣ４１Ｈ　　下位８ビット

ＥＣ４２Ｈ　　上位８ビット（最上位ビットは符号用のため７ビット）

ＥＡＢＤＨ＝０４Ｈ（単精度型実数）

ＥＣ４１Ｈ　　仮数部下位８ビット

ＥＣ４２Ｈ　　仮数部中位８ビット

ＥＣ４３Ｈ　　仮数部上位８ビット（最上位ビットは符号用）

ＥＣ４４Ｈ　　指数部８ビット

ＥＡＢＤＨ＝０８Ｈ（倍精度型実数）

ＥＣ３ＤＨ　　仮数部最下位８ビット

ＥＣ３ＥＨ　　仮数部中位８ビット

ＥＣ３ＦＨ　　仮数部中位８ビット

ＥＣ４０Ｈ　　仮数部中位８ビット

ＥＣ４１Ｈ　　仮数部中位８ビット

ＥＣ４２Ｈ　　仮数部中位８ビット

ＥＣ４３Ｈ　　仮数部最上位８ビット（最上位ビットは符号用）

ＥＣ４４Ｈ　　指数部８ビット

９７ＥＡＨ　リムーブ処理を行う．

アドレス	９７ＥＡＨ
機　能	指定デバイスのリムーブ処理を行う．
レジスタ	Ａ，Ｆ

解　説　アキュムレータの値で示されるデバイス（１－９を参照）がディスクで，かつそのデバイス上にファイルが１つも開いていない場合に，ＦＡＴの更新を行うサブルーチンです．指定デバイスがディスクでない場合には何もせずに戻ります．また，このサブルーチンはＮ－ＤＩＳＫ ＢＡＳＩＣのＲＥＭＯＶＥ文と異なり開いているファイルをクローズしません．

指定デバイスが非マウント状態にあったときには，エラーは発生せずにリムーブ処理を行わずに戻ります．またＦＡＴの更新（書き込み）が行われる際には，ライトエラーが３回に達すると Bad allocation table エラーが発生し，エラーの数が３回未満ならばＣＲＴ上に X copies of allocation bad on drive Y のメッセージを出力し，処理を続行します．

サンプル

```
                 ;
                 ; --- remove all drives
                 ;
                 ;       sample of subroutine from 97EAH
                 ;
                         ORG   0B900H
                 ;
B900 2133B9      RMVALL:LD    HL,MESS
B903 CD2AB9              CALL  DSPMES
                 ;
B906 0E00                LD    C,0            ; top of drive
                 ;
B908 79          RMVL:   LD    A,C
B909 CDEA97              CALL  97EAH          ; remove drive
                 ;
B90C 214EB9              LD    HL,RMVDR1
B90F CD2AB9              CALL  DSPMES
                 ;
B912 C5                  PUSH  BC             ; print drive
B913 69                  LD    L,C
B914 2600                LD    H,0
B916 23                  INC   HL
B917 CDC228              CALL  28C2H          ; display integer
B91A C1                  POP   BC
                 ;
B91B 2157B9              LD    HL,RMVDR2
B91E CD2AB9              CALL  DSPMES
                 ;
B921 0C                  INC   C
B922 3A7DEC              LD    A,(0EC7DH)     ; maximum drive number
B925 3D                  DEC   A
B926 B9                  CP    C
B927 30DF                JR    NC,RMVL
                 ;
B929 C9                  RET
```

```
                  ;
B92A 7E         DSPMES:LD    A,(HL)              ; display message subroutine
B92B A7                AND   A
B92C C8                RET   Z
B92D CD0D3E            CALL  3E0DH
B930 23                INC   HL
B931 18F7              JR    DSPMES
                  ;
B933 0D0A2D2D MESS:    DEFM  0DH,0AH,'---REMOVE ALL DRIVE---',0DH,0AH,0
B937 2D52454D
B93B 4F564520
B93F 414C4C20
B943 44524956
B947 452D2D2D
B94B 0D0A00
                  ;
B94E 0D0A4472 RMVDR1:DEFM 0DH,0AH,'Drive ',0
B952 69766520
B956 00
B957 2072656D RMVDR2:DEFM ' removed.',0DH,0AH,0
B95B 6F766564
B95F 2E0D0A00
                  ;
B963                   END
```

```
a=&hb900:call a

---REMOVE ALL DRIVE---

Drive 1 removed.

Drive 2 removed.

Drive 3 removed.

Drive 4 removed.
Ok
```

９８２１H　マウント処理を行う．

アドレス	９８２１H
機　能	指定デバイスのマウント処理を行う．
レジスタ	A，F

解　説　アキュムレータの値で示されるデバイス（１－９を参照）がディスクである場合に，ＦＡＴの読み出し・ＩＤセクタの読み出し・空きクラスタの算出などのマウント処理を行うサブルーチンで，Ｎ－ＤＩＳＫ ＢＡＳＩＣのＭＯＵＮＴ文と同等の処理を行うものです．リターン時にはアキュムレータに空きクラスタ数が格納されています．与えられたデバイスナンバ・がディスク以外のデバイスならば，このサブルーチンは何もせずに戻ります．

また，指定デバイス（ドライブ）がすでにマウントされていた場合にはＤＩＳＫ already mounted エラーが発生し，ＦＡＴを読み出す際にリードエラーが３回に達すると Bad allocation table エラーが発生します．エラーの数が３回未満ならばＣＲＴ上に X copies of allocation bad on drive Y のメッセージを出力し，処理を続行します．

サンプル

```
                ;
                ; --- mount all drive and get free clusters
                ;
                ;       sample of subroutine from 9821H
                ;
                        ORG   0B900H
                ;
B900 2143B9     MNTALL:LD    HL,MESS
B903 CD3AB9            CALL  DSPMES
                ;
B906 0E00              LD    C,0               ; top of drive
                ;
B908 79         MNTL:  LD    A,C
B909 CD2198            CALL  9821H             ; mount a disk
                ;
B90C F5                PUSH  AF
B90D 215DB9            LD    HL,MNTDR1         ; message for mounted
B910 CD3AB9            CALL  DSPMES
                ;
B913 79                LD    A,C
B914 3C                INC   A
B915 CD31B9            CALL  DSPACC            ; print drive number
                ;
B918 2166B9            LD    HL,MNTDR2
B91B CD3AB9            CALL  DSPMES
                ;
B91E F1                POP   AF
B91F CD31B9            CALL  DSPACC
                ;
B922 2170B9            LD    HL,FRCLST         ; message for free clusters
B925 CD3AB9            CALL  DSPMES
                ;
B928 0C                INC   C
```

```
B929 3A7DEC          LD   A,(0EC7DH)        ; maximum drive number
B92C 3D              DEC  A
B92D B9              CP   C
B92E 30D8            JR   NC,MNTL
              ;
B930 C9              RET
              ;
B931 C5       DSPACC:PUSH BC                ; display accumulator subroutine
B932 6F              LD   L,A
B933 2600            LD   H,0
B935 CDC228          CALL 28C2H             ; display integer
B938 C1              POP  BC
B939 C9              RET
              ;
B93A 7E       DSPMES:LD   A,(HL)            ; display message subroutine
B93B A7              AND  A
B93C C8              RET  Z
B93D CD0D3E          CALL 3E0DH
B940 23              INC  HL
B941 18F7            JR   DSPMES
              ;
B943 0D0A2D2D MESS:  DEFM 0DH,0AH,'---MOUNT ALL DRIVE---',0DH,0AH,0
B947 2D4D4F55
B94B 4E542041
B94F 4C4C2044
B953 52495645
B957 2D2D2D0D
B95B 0A00
              ;
B95D 0D0A4472 MNTDR1:DEFM 0DH,0AH,'Drive ',0
B961 69766520
B965 00
B966 206D6F75 MNTDR2:DEFM ' mounted,',0
B96A 6E746564
B96E 2C00
              ;
B970 20636C75 FRCLST:DEFM ' clusters free.',0DH,0AH,0
B974 73746572
B978 73206672
B97C 65652E0D
B980 0A00
              ;
B982                 END
```

```
a=&hb900:call a

---MOUNT ALL DRIVE---

Drive 1 mounted,113 clusters free.

Drive 2 mounted,138 clusters free.

Drive 3 mounted,103 clusters free.

Drive 4 mounted,96 clusters free.
Ok
call a

---MOUNT ALL DRIVE---
Disk already mounted
Ok
```

９８８６Ｈ　ＦＡＴコピーのアドレスをセットする．

アドレス	９８８６Ｈ
機　　能	ＦＡＴコピーのアドレスをセットする．
レジスタ	Ｄ，Ｅ，Ｈ，Ｌ

解　　説　ドライブのマウント時に，メモリ上にコピーされるＦＡＴの先頭アドレスを求め，値をＤＥレジスタペアに与えます．この場合，ＥＣ８６Ｈ，８７Ｈ番地には指定ドライブのドライブバッファ・アドレス（１－８を参照）がセットされている必要がありますが，これには９９５３Ｈ番地からのサブルーチンを利用するのが最適です．

９８ＤＡＨ　クラスタ数からセクタ数への変換を行う.

アドレス　９８ＤＡＨ

機　　能　クラスタ数からセクタ数への変換を行う.

レジスタ　A, F, D, E, H, L

解　　説　Ｃレジスタに与えられたクラスタ数をセクタ数に変換し，ＤＥレジスタペアにその値を返すサブルーチンです．このサブルーチンでは計算をワークエリアの値を参照して行うため，ドライブのタイプに応じてワークエリアに値をセットしておく必要があります．これには，９９５３Ｈ番地からのサブルーチンを利用するのが最適です．

サンプル

```
                ;
                ; --- clustes to sector converter
                ;
                ;       sample of subroutine from 98DAH
                ;
                        ORG    0B900H
                ;
B900 2145B9     CLSSEC:LD     HL,CLUST            ; input cluster number
B903 CD3CB9            CALL   DSPMES
B906 CDC85F            CALL   5FC8H               ; line input subroutine
B909 D8                RET    C                   ; stop key then
                ;
B90A 23                INC    HL
B90B E5                PUSH   HL
                ;
B90C CDBC26            CALL   26BCH               ; character to binary
B90F CDA021            CALL   21A0H               ; convert to integer
B912 4D                LD     C,L
B913 CDDA98            CALL   98DAH               ; convert cluster to sector
                ;
B916 2159B9            LD     HL,MESS1
B919 CD3CB9            CALL   DSPMES
                ;
B91C E1                POP    HL
B91D CD3CB9            CALL   DSPMES              ; cluster display
                ;
B920 2162B9            LD     HL,MESS2
B923 CD3CB9            CALL   DSPMES
                ;
B926 13                INC    DE
B927 D5                PUSH   DE
B928 EB                EX     DE,HL
B929 CDC228            CALL   28C2H               ; sector display
                ;
B92C 216EB9            LD     HL,MESS3
B92F CD3CB9            CALL   DSPMES
B932 D1                POP    DE
                ;
B933 210700            LD     HL,7                ; sector +8 display
B936 19                ADD    HL,DE
B937 CDC228            CALL   28C2H
                ;
B93A 18C4              JR     CLSSEC
                ;
                ; subroutine display message
                ;
B93C 7E         DSPMES:LD     A,(HL)
```

```
B93D A7              AND  A
B93E C8              RET  Z
B93F CD0D3E          CALL 3E0DH          ; one character output to CRT
B942 23              INC  HL
B943 18F7            JR   DSPMES
                ;
                ; message area
                ;
B945 0D0A436C CLUST: DEFM 0DH,0AH,'Cluster number ? ',0
B949 75737465
B94D 72206E75
B951 6D626572
B955 203F2000
                ;
B959 436C7573 MESS1: DEFM 'Cluster ',0
B95D 74657220
B961 00
B962 202D3E20 MESS2: DEFM ' -> Sector ',0
B966 53656374
B96A 6F722000
B96E 20746F20 MESS3: DEFM ' to ',0
B972 00
                ;
B973                 END
```

```
a=&hb900:call a
```

```
Cluster number ? 0
Cluster 0 -> Sector 1 to 8
Cluster number ? 79
Cluster 79 -> Sector 633 to 640
Cluster number ? 159
Cluster 159 -> Sector 1273 to 1280
Cluster number ?
Ok
```

98EAH　ディレクトリよりファイル名をサーチする.

アドレス　98EAH

機　　能　ディレクトリよりファイル名をサーチする.

レジスタ　A, F, B, C, D, E, H, L

解　　説　ディレクトリより指定されたファイル名を検索し，その位置を与えるサブルーチンです．ファイル名の指定は次のように行います．

EC8FH～EC94H　　ファイル名の前半6バイト

EC95H～EC97H　　拡張子3バイト

ここで，ファイル名の前半が6バイトに満たない場合は残りをスペースで埋めることが必要です．拡張子のない場合も同様です．

また，サブルーチンから戻ったときの情報は以下の様になります．

Zフラグ　………ファイル名が見つかったか否かの情報を与えます．見つかったときにはセットされます．

EC8AH，8BH，HLレジスタペア

バッファ内のファイル名の先頭アドレスを与えます．

EC8CH………ディレクトリ・セクタ内において，何番目のファイル名かを与えます．実際には16からその値を引いたものを与えます．もし，ファイルの見つからなかった場合には第7ビットがセットされます．

EC8DH………ファイル名の見つかったセクタの番号を与えます．

サンプル

```
                ;
                ; --- file name search
                ;
                ;       sample of subroutine from 98EAH
                ;
                       ORG  0B900H
                ;
B900 2163B9     FNSEAR:LD   HL,FILENM           ; input file name
B903 CD5AB9            CALL DSPMES
B906 CDC85F            CALL 5FC8H               ; line input subroutine
B909 D8                RET  C                   ; stop key return
                ;
B90A 3622              LD   (HL),'"'
B90C CD8C46            CALL 468CH
B90F A7                AND  A
B910 FA00B9            JP   M,FNSEAR            ; non disk file
                ;
B913 0600              LD   B,0                 ; top of drive
                ;
B915 78         FNSEAL:LD   A,B

                            .
```

```
B916 C5                 PUSH BC
B917 CD5899             CALL 9958H              ; set disk identification
                ;
B91A CDEA98             CALL 98EAH              ; directory search
                ;
B91D C1                 POP  BC
B91E 2810               JR   Z,FOUND
B920 04                 INC  B
B921 3A7DEC             LD   A,(0EC7DH)         ; maximum drive number
B924 3D                 DEC  A
B925 B8                 CP   B
B926 30ED               JR   NC,FNSEAL          ; all drive check
                ;
B928 218DB9             LD   HL,NFOUND          ; not found
B92B CD5AB9             CALL DSPMES
                ;
B92E 181E               JR   FOUNDJ
                ;
B930 E5         FOUND:  PUSH HL
B931 2172B9             LD   HL,FILMES
B934 CD5AB9             CALL DSPMES
B937 E1                 POP  HL
B938 0E09               LD   C,9                ; print file name
B93A 7E         FOUNDL:LD    A,(HL)
B93B CD0D3E             CALL 3E0DH
B93E 23                 INC  HL
B93F 0D                 DEC  C
B940 20F8               JR   NZ,FOUNDL
                ;
B942 217FB9             LD   HL,DRVMES
B945 CD5AB9             CALL DSPMES
                ;
B948 78                 LD   A,B                ; print drive number
B949 C631               ADD  A,'1'              ; convert to ascii
B94B CD0D3E             CALL 3E0DH
B94E 3E0D       FOUNDJ:LD    A,0DH              ; line feed
B950 CD0D3E             CALL 3E0DH
B953 3E0A               LD   A,0AH
B955 CD0D3E             CALL 3E0DH
                ;
B958 18A6               JR   FNSEAR
                ;
B95A 7E         DSPMES:LD    A,(HL)
B95B A7                 AND  A
B95C C8                 RET  Z
B95D CD0D3E             CALL 3E0DH
B960 23                 INC  HL
B961 18F7               JR   DSPMES
                ;
B963 0D0A4669 FILENM:DEFM 0DH,0AH,'File name ? ',0
B967 6C65206E
B96B 616D6520
B96F 3F2000
B972 0D0A4669 FILMES:DEFM 0DH,0AH,'File name:',0
B976 6C65206E
B97A 616D653A
B97E 00
B97F 20697320 DRVMES:DEFM ' is in drive ',0
B983 696E2064
B987 72697665
B98B 2000
                ;
B98D 0D0A4E6F NFOUND:DEFM 0DH,0AH,'Not found !',7,0
B991 7420666F
B995 756E6420
B999 210700
                ;
B99C                    END
```

```
a=&hb900:call a

File name ? backupn88
```

```
File name:backupn88 is in drive 1

File name ? formatn88

Not found !

File name ?
Ok
```

９９５３Ｈ　ディスクの諸元，ドライブバッファのアドレスをセットする．

アドレス	９９５３Ｈ
機　　能	ディスクの諸元，ドライブバッファのアドレスをセットする．
レジスタ	Ａ，Ｆ，Ｄ，Ｅ，Ｈ，Ｌ

解　　説　アキュムレータの値で示されるドライブの各種情報をワークエリアにコピーし，またそのドライブのドライブバッファのアドレスを求め，ＥＣ８８Ｈ，８９Ｈ両番地にセットします．指定ドライブの各種情報については４７４２Ｈ番地からのサブルーチンに準じますが，このサブルーチンでは他にドライブバッファ（１－８参照）のアドレスを求め，その値をワークエリアにセットします．また，リターン時には指定ドライブ上にファイルを開いているか否かの情報を以下のように持っています．

アキュムレータの値が００Ｈ以外またはＺフラグがリセット
………ファイルがＯＰＥＮ

アキュムレータの値が００ＨまたはＺフラグがセット
………ファイルがＣＬＯＳＥ

アキュムレータの値が００Ｈ以下，または接続されているドライブの数を越えた場合には Bad drive number エラーが発生します．

サンプル

```
                   ;
                   ; --- print DSKF function and status
                   ;
                   ;       sample of subroutine from 9953H
                   ;
                          ORG   0B900H
                   ;
B900 21E7B9        PRSTAT:LD    HL,DRIVE          ; input drive number
B903 CDB9B9               CALL  DSPMES
B906 CDC85F               CALL  5FC8H             ; line input subroutine
B909 D8                   RET   C                 ; stop key return
                   ;
B90A 23                   INC   HL
B90B CDBC26               CALL  26BCH             ; character to binary code
B90E CDA021               CALL  21A0H             ; convert FAC to integer
                   ;
B911 7D                   LD    A,L
B912 CD5399               CALL  9953H             ; set drive ID
                   ;
B915 21F9B9               LD    HL,DDRIVE         ; display drive number
B918 CDB9B9               CALL  DSPMES
B91B 3A85EC               LD    A,(0EC85H)        ; drive number
B91E 3C                   INC   A
B91F CDD3B9               CALL  DSPACC
B922 3E14                 LD    A,20
B924 3287EF               LD    (0EF87H),A        ; cursor X
```

```
                    ;
B927 2A86EC              LD    HL,(0EC86H)         ; drive buffer address
B92A 7E                  LD    A,(HL)
B92B 3C                  INC   A
B92C CCC2B9              CALL  Z,NOTOPN            ; file not open
B92F 21A8BA              LD    HL,DSPOPN
B932 CDB9B9              CALL  DSPMES
                    ;
B935 11A5EC              LD    DE,0ECA5H           ; set top of data
                    ;
B938 2100BA              LD    HL,MAXTRK           ; maximum track number
B93B CDB9B9              CALL  DSPMES
B93E 1A                  LD    A,(DE)
B93F CDD3B9              CALL  DSPACC
B942 CDDCB9              CALL  CRLF
B945 13                  INC   DE
                    ;
B946 1A                  LD    A,(DE)              ; sector number par track
B947 CDD3B9              CALL  DSPACC
B94A 2119BA              LD    HL,SECTRK
B94D CDB9B9              CALL  DSPMES
B950 13                  INC   DE
                    ;
B951 1A                  LD    A,(DE)              ; surface
B952 B7                  OR    A
B953 CCC7B9              CALL  Z,SNGSUR
B956 C4CCB9              CALL  NZ,DBLSUR
B959 21C9BA              LD    HL,SURFAC
B95C CDB9B9              CALL  DSPMES
B95F 13                  INC   DE
                    ;
B960 1A                  LD    A,(DE)              ; cluster number par track
B961 CDD3B9              CALL  DSPACC
B964 212ABA              LD    HL,CLSTRK
B967 CDB9B9              CALL  DSPMES
B96A 13                  INC   DE
                    ;
B96B 1A                  LD    A,(DE)              ; cluster number par volume
B96C CDD3B9              CALL  DSPACC
B96F 213CBA              LD    HL,CLSVOL
B972 CDB9B9              CALL  DSPMES
B975 13                  INC   DE
                    ;
B976 214FBA              LD    HL,DIRTRK           ; display directory track
B979 CDB9B9              CALL  DSPMES
B97C 1A                  LD    A,(DE)
B97D CDD3B9              CALL  DSPACC
B980 CDDCB9              CALL  CRLF
B983 13                  INC   DE
                    ;
B984 1A                  LD    A,(DE)              ; sector number par cluster
B985 CDD3B9              CALL  DSPACC
B988 2163BA              LD    HL,SECCLS
B98B CDB9B9              CALL  DSPMES
B98E 13                  INC   DE
                    ;
B98F 2176BA              LD    HL,FATTOP           ; display FAT sector
B992 CDB9B9              CALL  DSPMES
B995 1A                  LD    A,(DE)
B996 CDD3B9              CALL  DSPACC
B999 2185BA              LD    HL,FATEND
B99C CDB9B9              CALL  DSPMES
B99F 13                  INC   DE
B9A0 1A                  LD    A,(DE)
B9A1 CDD3B9              CALL  DSPACC
B9A4 CDDCB9              CALL  CRLF
B9A7 13                  INC   DE
B9A8 13                  INC   DE
                    ;
B9A9 218ABA              LD    HL,ATRSEC           ; display ID sector
B9AC CDB9B9              CALL  DSPMES
B9AF 1A                  LD    A,(DE)
B9B0 CDD3B9              CALL  DSPACC
B9B3 CDDCB9              CALL  CRLF
```

```
                  ;
B9B6 C300B9              JP    PRSTAT
                  ;
                  ; subroutine display message
                  ;
B9B9 7E           DSPMES:LD    A,(HL)
B9BA A7                  AND   A
B9BB C8                  RET   Z
B9BC CD0D3E              CALL  3E0DH              ; one character output to CRT
B9BF 23                  INC   HL
B9C0 18F7                JR    DSPMES
                  ;
                  ; file not open
                  ;
B9C2 21A4BA       NOTOPN:LD    HL,NOFILE
B9C5 1808                JR    DSPSUR
                  ;
                  ; single surface
                  ;
B9C7 21BBBA       SNGSUR:LD    HL,SINGLE
B9CA 1803                JR    DSPSUR
                  ;
                  ; double surface
                  ;
B9CC 21C2BA       DBLSUR:LD    HL,DOUBLE
B9CF CDB9B9       DSPSUR:CALL  DSPMES
B9D2 C9                  RET
                  ;
                  ; subroutine display accumlator
                  ;
B9D3 D5           DSPACC:PUSH  DE
B9D4 6F                  LD    L,A
B9D5 2600                LD    H,0
B9D7 CDC228              CALL  28C2H              ; display HL
B9DA D1                  POP   DE
B9DB C9                  RET
                  ;
                  ; subroutine line feed
                  ;
B9DC 3E0D         CRLF:  LD    A,0DH
B9DE CD0D3E              CALL  3E0DH
B9E1 3E0A                LD    A,0AH
B9E3 CD0D3E              CALL  3E0DH
B9E6 C9                  RET
                  ;
                  ; message area
                  ;
B9E7 0D0A4472 DRIVE: DEFM 0DH,0AH,'Drive number ? ',0
B9EB 69766520
B9EF 6E756D62
B9F3 6572203F
B9F7 2000
                  ;
B9F9 44524956 DDRIVE:DEFM 'DRIVE:',0
B9FD 453A00
                  ;
BA00 4D415849 MAXTRK:DEFM 'MAXIMUM TRACK NUMBER IS ',0
BA04 4D554D20
BA08 54524143
BA0C 4B204E55
BA10 4D424552
BA14 20495320
BA18 00
                  ;
BA19 20534543 SECTRK:DEFM ' SECTORS/TRACK',0DH,0AH,0
BA1D 544F5253
BA21 2F545241
BA25 434B0D0A
BA29 00
                  ;
BA2A 20434C55 CLSTRK:DEFM ' CLUSTERS/TRACK',0DH,0AH,0
BA2E 53544552
BA32 532F5452
BA36 41434B0D
```

```
BA3A 0A00
                  ;
BA3C 20434C55 CLSVOL:DEFM ' CLUSTERS/VOLUME',0DH,0AH,0
BA40 53544552
BA44 532F564F
BA48 4C554D45
BA4C 0D0A00
                  ;
BA4F 44495245 DIRTRK:DEFM 'DIRECTORY IS TRACK ',0
BA53 43544F52
BA57 59204953
BA5B 20545241
BA5F 434B2000
                  ;
BA63 20534543 SECCLS:DEFM ' SECTORS/CLUSTER',0DH,0AH,0
BA67 544F5253
BA6B 2F434C55
BA6F 53544552
BA73 0D0A00
                  ;
BA76 46415420 FATTOP:DEFM 'FAT IS SECTOR ',0
BA7A 49532053
BA7E 4543544F
BA82 522000
                  ;
BA85 20544F20 FATEND:DEFM ' TO ',0
BA89 00
                  ;
BA8A 4449534B ATRSEC:DEFM 'DISK ATTRIBUTE IS SECTOR ',0
BA8E 20415454
BA92 52494255
BA96 54452049
BA9A 53205345
BA9E 43544F52
BAA2 2000
                  ;
BAA4 4E4F2000 NOFILE:DEFM 'NO ',0
                  ;
BAA8 46494C45 DSPOPN:DEFM 'FILE IS OPENING.',0DH,0AH,0
BAAC 20495320
BAB0 4F50454E
BAB4 494E472E
BAB8 0D0A00
                  ;
BABB 53494E47 SINGLE:DEFM 'SINGLE',0
BABF 4C4500
                  ;
BAC2 444F5542 DOUBLE:DEFM 'DOUBLE',0
BAC6 4C4500
                  ;
BAC9 20535552 SURFAC:DEFM ' SURFACE ',0DH,0AH,0
BACD 46414345
BAD1 200D0A00
                  ;
BAD5                    END


a=&hb900:call a

Drive number ? 1
DRIVE:1              NO FILE IS OPENING.
MAXIMUM TRACK NUMBER IS 39
16 SECTORS/TRACK
DOUBLE SURFACE
2 CLUSTERS/TRACK
160 CLUSTERS/VOLUME
DIRECTORY IS TRACK 18
8 SECTORS/CLUSTER
FAT IS SECTOR 14 TO 16
DISK ATTRIBUTE IS SECTOR 13

Drive number ?
Ok
```

９９９７H　ディスク属性のあるトラック，セクタ・ナンバをセットする．

アドレス	９９９７H
機　能	ディスク属性のあるトラック，セクタ・ナンバをセットする．
レジスタ	B，C

解　説　ディスク属性のあるトラック(ディレクトリ・トラック)，セクタ(IDセクタ）を求め，トラック・ナンバを論理トラックでBレジスタに，セクタ・ナンバをCレジスタにそれぞれ与えます．

このサブルーチンでは計算をワークエリアの値を参照して行うため，ドライブのタイプに応じてワークエリアに値をセットしておくことが必要です．これには，９９５３H番地からのサブルーチンを利用するのが最適です．

サンプル

```
                ;
                ; --- get disk attribute
                ;
                ;       sample of subroutine from 9997H
                ;
                        ORG   0B900H
                ;
B900 1601       GETATR:LD    D,1                ; top of drive number
                ;
B902 7A         GETATL:LD    A,D
B903 D5                PUSH  DE
B904 CD5399            CALL  9953H              ; set disk identification
                ;
B907 CD9799            CALL  9997H              ; set ID track,sector
B90A 217AB9            LD    HL,BUFFER          ; reading buffer
B90D 3E01              LD    A,1
B90F B7                OR    A                  ; clear carry,zero flag
B910 CD9A36            CALL  369AH              ; read ID sector
B913 D1                POP   DE
B914 DA2BA7            JP    C,0A72BH           ; disk I/O error
                ;
B917 D5                PUSH  DE
B918 2162B9            LD    HL,DRIVE           ; print drive number
B91B CD59B9            CALL  DSPMES
B91E 7A                LD    A,D
B91F C630              ADD   A,'0'              ; convert to ascii
B921 CD0D3E            CALL  3E0DH
                ;
B924 216BB9            LD    HL,DRIVE2
B927 CD59B9            CALL  DSPMES
                ;
B92A 217AB9            LD    HL,BUFFER
B92D 7E                LD    A,(HL)             ; get attribute
B92E CB77              BIT   6,A                ; check R
B930 C447B9            CALL  NZ,DISPR
B933 CC54B9            CALL  Z,DISPN
B936 CB67              BIT   4,A                ; check P
B938 C44FB9            CALL  NZ,DISPP
B93B CC54B9            CALL  Z,DISPN
                ;
B93E D1                POP   DE
B93F 14                INC   D
```

```
B940 3A7DEC          LD   A,(0EC7DH)          ; maximum drive number
B943 BA              CP   D
B944 30BC            JR   NC,GETATL
                ;
B946 C9              RET
                ;
B947 F5         DISPR: PUSH AF                ; print R attribute
B948 3E52              LD   A,'R'
B94A CD0D3E     DISPJ: CALL 3E0DH
B94D F1                POP  AF
B94E C9                RET
                ;
B94F F5         DISPP: PUSH AF                ; print P attribute
B950 3E50              LD   A,'P'
B952 18F6              JR   DISPJ
                ;
B954 F5         DISPN: PUSH AF                ; print non attribute
B955 3E2D              LD   A,'-'
B957 18F1              JR   DISPJ
                ;
B959 7E         DSPMES:LD   A,(HL)            ; display message subroutine
B95A A7                AND  A
B95B C8                RET  Z
B95C CD0D3E            CALL 3E0DH
B95F 23                INC  HL
B960 18F7              JR   DSPMES
                ;
B962 0D0A4472 DRIVE: DEFM 0DH,0AH,'Drive ',0
B966 69766520
B96A 00
B96B 20617474 DRIVE2:DEFM ' attribute is ',0
B96F 72696275
B973 74652069
B977 732000
                ;
B97A            BUFFER:DEFS 256
                ;
BA7A                   END
```

```
a=&hb900:call a

Drive 1 attribute is RP
Drive 2 attribute is --
Ok
```

９９Ａ１Ｈ　ディレクトリトラックをセットする．

アドレス	９９Ａ１Ｈ
機　　能	ディレクトリトラックをセットする．
レジスタ	Ｂ

解　　説　ディレクトリのあるトラック(ＩＤセクタ，ＦＡＴを含むトラック)を求め，値を論理トラックでＢレジスタに与えます．

このサブルーチンでは計算をワークエリアの値を参照して行うため，ドライブのタイプに応じてワークエリアに値をセットしておくことが必要です．
これには，９９５３Ｈ番地からのサブルーチンを利用するのが最適です．

9 9 E F H　ネーム・バッファの内容を交換する．

アドレス	9 9 E F H
機　能	ファイルネーム・バッファの内容を交換する．
レジスタ	A，F，B，C，D，E，H，L

解　説　ＥＣ８ＦＨ番地とＥＣ９８Ｈ番地のそれぞれから始まる９バイトづつのファイルネーム・バッファの内容を交換するサブルーチンです．N_{88}－ＤＩＳＫ ＢＡＳＩＣでは，この２番目(ＥＣ９８Ｈ～）のバッファをＮＡＭＥ文において使用されるのみですが，ファイル名の変更などには便利なサブルーチンです．

サンプル

```
                 ;
                 ; file name change program
                 ;
                 ;       sample of subroutine from 99EFH
                 ;
                        ORG  0B900H
                 ;
B900 2175B9      CHGFNM:LD   HL,FILE1             ; input exist file name
B903 CD6CB9             CALL DSPMES
B906 CDC85F             CALL 5FC8H
B909 D8                 RET  C                    ; if stop key return
                 ;
B90A 3622               LD   (HL),'"'             ; double quatation
B90C CD8C46             CALL 468CH                ; processing of file descriptor
B90F B7                 OR   A
B910 FA54B9             JP   M,BADNAM             ; non disk file
B913 D5                 PUSH DE
                 ;
B914 CDEF99             CALL 99EFH                ; exchange file name buffer
                 ;
B917 218AB9             LD   HL,FILE2             ; input new file name
B91A CD6CB9             CALL DSPMES
B91D CDC85F             CALL 5FC8H
                 ;
B920 3622               LD   (HL),'"'
B922 CD8C46             CALL 468CH
B925 F1                 POP  AF
B926 BA                 CP   D
B927 2023               JR   NZ,RENAME
                 ;
B929 B7                 OR   A
B92A FA54B9             JP   M,BADNAM             ; non disk file
                 ;
B92D CD5899             CALL 9958H                ; set disk identification
B930 CDEA98             CALL 98EAH                ; file name serach from directory
B933 2827               JR   Z,EXIST              ; file already exists
                 ;
B935 CDEF99             CALL 99EFH
B938 CDEA98             CALL 98EAH
B93B 2027               JR   NZ,NFOUND            ; file not found
B93D 1198EC             LD   DE,0EC98H            ; file name buffer 2
B940 CDF599             CALL 99F5H
B943 3A8DEC             LD   A,(0EC8DH)           ; sector number of file
B946 57                 LD   D,A
B947 CD40A3             CALL 0A340H
                 ;
B94A 18B4               JR   CHGFNM
```

```
                ;
B94C 219BB9     RENAME:LD   HL,ERR73
B94F CD6CB9            CALL DSPMES
B952 18AC              JR   CHGFNM
                ;
B954 21B0B9     BADNAM:LD   HL,ERR56
B957 CD6CB9            CALL DSPMES
B95A 18A4              JR   CHGFNM
                ;
B95C 21BFB9     EXIST: LD   HL,ERR65
B95F CD6CB9            CALL DSPMES
B962 189C              JR   CHGFNM
                ;
B964 21D4B9     NFOUND:LD   HL,ERR53
B967 CD6CB9            CALL DSPMES
B96A 1894              JR   CHGFNM
                ;
                ; subroutine display message
                ;
B96C 7E         DSPMES:LD   A,(HL)
B96D A7                AND  A
B96E C8                RET  Z
B96F CD0D3E            CALL 3E0DH
B972 23                INC  HL
B973 18F7              JR   DSPMES
                ;
                ; message area
                ;
B975 0D0A4578 FILE1: DEFM 0DH,0AH,'Exist file name ? ',0
B979 69737420
B97D 66696C65
B981 206E616D
B985 65203F20
B989 00
                ;
B98A 4E657720 FILE2: DEFM 'New file name ? ',0
B98E 66696C65
B992 206E616D
B996 65203F20
B99A 00
                ;
B99B 52656E61 ERR73: DEFM 'Rename across disks',7,0
B99F 6D652061
B9A3 63726F73
B9A7 73206469
B9AB 736B7307
B9AF 00
                ;
B9B0 42616420 ERR56: DEFM 'Bad file name',7,0
B9B4 66696C65
B9B8 206E616D
B9BC 650700
                ;
B9BF 46696C65 ERR65: DEFM 'File already exists',7,0
B9C3 20616C72
B9C7 65616479
B9CB 20657869
B9CF 73747307
B9D3 00
                ;
B9D4 46696C65 ERR53: DEFM 'File not found',7,0
B9D8 206E6F74
B9DC 20666F75
B9E0 6E640700
                ;
B9E4                   END
```

```
files
backup.n88 2   file   dat 1   setinf.n88 1   xfiles.n88 1   sysgen.    1
dummy1     1   dummy2     1   dummy3     1   sample dat 1   dummy4     1
```

```
machin*e    2     sample*bin 1     dummy5      1     This m ess 1     dummy6      1
S     .     1
Ok
a=&hb900:call a


Exist file name ? scrn:S
Bad file name
Exist file name ? SS
New file name ? SSS
File not found
Exist file name ? S
New file name ? 2:S
Rename across disks
Exist file name ? S
New file name ? SS

Exist file name ?
Ok
files
backup.n88 2     file   dat 1     setinf.n88 1     xfiles.n88 1     sysgen.     1
dummy1      1     dummy2      1     dummy3      1     sample dat 1     dummy4      1
machin*e    2     sample*bin 1     dummy5      1     This m ess 1     dummy6      1
SS          1
Ok
```

９ＢＢ６Ｈ　ディスクよりファイルを削除する．

アドレス	９ＢＢ６Ｈ
機　能	ディスクより指定ファイルを削除する．
レジスタ	Ａ，Ｆ

解　説　指定されるファイルをディスク上から削除するサブルーチンで，ＫＩＬＬ文のメイン処理を行うものです．ドライブとファイル名の指定は以下の様にして行ないます．

　　Ｄレジスタ………ドライブ・ナンバ（０～）
　　ＥＣ８ＦＨ～ＥＣ９７Ｈ
　　　　　………ファイル名

　指定ドライブ上に指定ファイルが存在しない場合にはFile not foundエラーが発生し，また指定ファイルがオープン状態にあるときにはFile already ＯＰＥＮエラーが発生します．

サンプル

```
                ;
                ; --- kill disk file
                ;
                ;       sample of subroutine from 9BB6H
                ;
                        ORG   0B900H
                ;
B900 2139B9     KILLFL:LD    HL,FILENM          ; input file name
B903 CD30B9            CALL  DSPMES
B906 CDC85F            CALL  5FC8H              ; line input from keyboard
B909 D8                RET   C                  ; stop key return
                ;
B90A 3622              LD    (HL),'"'
B90C CD8C46            CALL  468CH              ; processing of file descriptor
                ;
B90F CDB69B            CALL  9BB6H              ; kill file
                ;
B912 2148B9            LD    HL,MESS            ; message for killed
B915 CD30B9            CALL  DSPMES
B918 218FEC            LD    HL,0EC8FH          ; top of name buffer
B91B 0609              LD    B,9
B91D 7E         DLOOP: LD    A,(HL)
B91E CD0D3E            CALL  3E0DH
B921 23                INC   HL
B922 10F9              DJNZ  DLOOP
                ;
B924 3E0D              LD    A,0DH              ; line feed
B926 CD0D3E            CALL  3E0DH
B929 3E0A              LD    A,0AH
B92B CD0D3E            CALL  3E0DH
                ;
B92E 18D0              JR    KILLFL
                ;
B930 7E         DSPMES:LD    A,(HL)             ; display message subroutine
B931 A7                AND   A
B932 C8                RET   Z
B933 CD0D3E            CALL  3E0DH
B936 23                INC   HL
```

```
B937 18F7              JR    DSPMES
                 ;
B939 0D0A4669 FILENM:DEFM 0DH,0AH,'File name ? ',0
B93D 6C65206E
B941 616D6520
B945 3F2000
                 ;
B948 4B696C6C MESS:  DEFM 'Killed file:',0
B94C 65642066
B950 696C653A
B954 00
                 ;
B955                   END
```

```
a=&hb900:call a

File name ? 3:backup.n88
Killed file:backupn88

File name ? 3:format.n88
File not found
Ok
```

９Ｆ６９Ｈ　ファイル名を出力する．

アドレス	９Ｆ６９Ｈ
機　能	ファイル名を出力する．
レジスタ	Ａ，Ｆ，Ｄ，Ｈ，Ｌ

解　説　９バイトから成るメモリ上のファイル名をＣＲＴあるいはプリンタへ出力するサブルーチンで，ＦＩＬＥＳ，ＬＦＩＬＥＳ文で使用されているものです．出力先（ＣＲＴ，プリンタ）の指定は，Ｅ６４ＣＨが００ＨのときＣＲＴ，それ以外の値でプリンタがそれぞれ選択されます．

また，このサブルーチンではＨＬレジスタの指すアドレスから９バイトのデータを出力しますが，その次の１０バイト目も意味を持ち，この１０バイト目の値によって６文字目と７文字目の間に決まったキャラクタを出力します．１０バイト目と出力されるキャラクタの関係は以下の様になります．

００Ｈ………スペース（２０Ｈ）．

０１Ｈ………アスタリスク（２ＡＨ）．

８０Ｈ………ピリオド（２ＥＨ）．

これにより，ファイルの内容による区別が可能となります．

サンプル

```
                ;
                ; --- print all file
                ;
                ;       sample of subroutine from 9F69H
                ;
                        ORG   0B900H
                ;
B900 AF         DEVICE:XOR   A                  ; clear output flag
B901 324CE6             LD    (0E64CH),A
                ;
B904 218FB9     INPDRV:LD    HL,DRIVE           ; input drive
B907 CD5DB9             CALL  DSPMES
B90A CDC85F             CALL  5FC8H
B90D D8                 RET   C                 ; stop key return
                ;
B90E 23                 INC   HL
B90F CDBC26             CALL  26BCH             ; convert to binary
B912 CDA021             CALL  21A0H             ; convert to integer
                ;
B915 3A7DEC             LD    A,(0EC7DH)        ; maximum drive number
B918 BD                 CP    L
B919 38E9               JR    C,INPDRV
B91B 2D                 DEC   L
B91C 7D                 LD    A,L
B91D CD5899             CALL  9958H             ; set disk identification
                ;
B920 216DB9             LD    HL,DEVNAM         ; input device for output
B923 CD5DB9             CALL  DSPMES
B926 CDC85F             CALL  5FC8H             ; line input from keyboard
B929 D8                 RET   C
                ;
```

```
B92A 23                INC  HL
B92B 7E                LD   A,(HL)
B92C FE70              CP   'p'
B92E 2003              JR   NZ,DIR
                ;
B930 324CE6            LD   (0E64CH),A
B933 1601       DIR:   LD   D,1             ; top of directory
B935 CD27A3     DIRL:  CALL 0A327H          ; directory read
                ;
B938 1E10              LD   E,16            ; number of file a sector
                ;
B93A 7E         DSPL:  LD   A,(HL)
B93B B7                OR   A
B93C 280D              JR   Z,SKIP          ; killed file
                ;
B93E 3C                INC  A
B93F 2817              JR   Z,EXIT          ; end of directory
                ;
B941 E5                PUSH HL
B942 D5                PUSH DE
B943 CD699F            CALL 9F69H           ; output file name
B946 CD66B9            CALL CRLF
B949 D1                POP  DE
B94A E1                POP  HL
B94B 011000     SKIP:  LD   BC,16           ; next file
B94E 09                ADD  HL,BC
B94F 1D                DEC  E
B950 20E8              JR   NZ,DSPL
                ;
B952 14                INC  D               ; next sector
B953 7A                LD   A,D
B954 FE11              CP   17
B956 38DD              JR   C,DIRL
                ;
B958 CD66B9     EXIT:  CALL CRLF
B95B 18A3              JR   DEVICE
                ;
B95D 7E         DSPMES:LD   A,(HL)          ; display message subroutine
B95E A7                AND  A
B95F C8                RET  Z
B960 CD0D3E            CALL 3E0DH
B963 23                INC  HL
B964 18F7              JR   DSPMES
                ;
B966 3E0D       CRLF:  LD   A,0DH           ; line feed subroutine
B968 DF                RST  18H
B969 3E0A              LD   A,0AH
B96B DF                RST  18H
B96C C9                RET
                ;
B96D 4F757470 DEVNAM:DEFM 'Output to CRT(c) '
B971 75742074
B975 6F204352
B979 54286329
B97D 20
B97E 6F722070          DEFM 'or printer(p) ? ',0
B982 72696E74
B986 65722870
B98A 29203F20
B98E 00
                ;
B98F 0D0A4472 DRIVE: DEFM 0DH,0AH,'Drive number ? ',0
B993 69766520
B997 6E756D62
B99B 6572203F
B99F 2000
                ;
B9A1                   END
```

```
a=&hb900:call a

Drive number ? 1
```

```
Output to CRT(c) or printer(p) ? c
backup.n88
file   dat
setinf.n88
xfiles.n88
sysgen.
dummy1
dummy2
dummy3
sample dat
dummy4
machin*e
sample*bin
dummy5
This m ess
dummy6
SS    .


Drive number ?
Ok
```

Ａ２Ｆ７Ｈ　ＦＡＴの読み書きを行う．

アドレス	Ａ２Ｆ７Ｈ
機　能	ＦＡＴの読み書きを行う．
レジスタ	Ａ，Ｆ，Ｂ，Ｃ，Ｈ，Ｌ

解　説　ＥＣ８６Ｈ，８７Ｈ両番地のアドレスで示されるドライブバッファについて，Ｄレジスタの値で示されるＦＡＴ（File Allocation Table）を読み出し，またはディスクへのＦＡＴの書き込みを行うサブルーチンです．

ＦＡＴは普通３セクタにわたって同一内容のものが書き込まれており，このサブルーチンではＤレジスタにＦＡＴの番号（１～３）＋１３（８インチの場合は２３）の値を与えることでどのＦＡＴかを指定します．また，読み出しを行うか，書き込みを行うかはＥレジスタで指定（００Ｈで読み出し，それ以外で書き込み）します．エラー発生の場合，アキュムレータにＦＦＨがセットされます．

サンプル

```
               ;
               ; --- check equality of FAT
               ;
               ;       sample of subroutine from A2F7H
               ;
                       ORG   0B900H
               ;
B900 2170B9    CHKFAT:LD    HL,DRIVE           ; input drive number
B903 CD67B9           CALL  DSPMES
B906 CDC85F           CALL  5FC8H              ; line input subroutine
B909 D8               RET   C                  ; stop key return
               ;
B90A 23               INC   HL
B90B CDBC26           CALL  26BCH              ; convert to binary
B90E CDA021           CALL  21A0H              ; convert to integer
B911 7D               LD    A,L
               ;
B912 3D               DEC   A
B913 CD5899           CALL  9958H              ; set disk identification
               ;
B916 3AACEC           LD    A,(0ECACH)         ; top of FAT
B919 57               LD    D,A
               ;
B91A 1E00      CHKFTL:LD    E,0                ; FAT read
B91C CDF7A2           CALL  0A2F7H
               ;
B91F 3AACEC           LD    A,(0ECACH)
B922 BA               CP    D
B923 C447B9           CALL  NZ,COMP            ; compare
               ;
B926 D5               PUSH  DE
B927 CD8698           CALL  9886H              ; FAT top to DE
B92A 21A8B9           LD    HL,FATBFR          ; FAT buffer address
B92D EB               EX    DE,HL
B92E 3AA9EC           LD    A,(0ECA9H)         ; cluster a volume
B931 4F               LD    C,A
B932 0600             LD    B,0
               ;
B934 EDB0             LDIR                     ; transfer FAT to buffer
```

```
                  ;
B936 D1                 POP  DE
B937 14                 INC  D
B938 3AADEC             LD   A,(0ECADH)          ; end of FAT
B93B 3C                 INC  A
B93C BA                 CP   D
B93D 20DB               JR   NZ,CHKFTL
                  ;
B93F 2182B9             LD   HL,CHKOK
B942 CD67B9             CALL DSPMES
                  ;
B945 18B9               JR   CHKFAT
                  ;
B947 D5           COMP: PUSH DE
B948 CD8698             CALL 9886H               ; compare two buffer
B94B 21A8B9             LD   HL,FATBFR
B94E EB                 EX   DE,HL
B94F 3AA9EC             LD   A,(0ECA9H)
B952 47                 LD   B,A
                  ;
B953 1A           COMPL: LD  A,(DE)
B954 BE                 CP   (HL)
B955 2006               JR   NZ,ERROR
B957 23                 INC  HL
B958 13                 INC  DE
B959 10F8               DJNZ COMPL
                  ;
B95B D1                 POP  DE
B95C C9                 RET
                  ;
B95D 2190B9       ERROR: LD  HL,ERRMES           ; FAT error
B960 CD67B9             CALL DSPMES
B963 D1                 POP  DE
B964 F1                 POP  AF                  ; cancel
                  ;
B965 1899               JR   CHKFAT
                  ;
B967 7E           DSPMES:LD  A,(HL)              ; dsplay message subroutine
B968 A7                 AND  A
B969 C8                 RET  Z
B96A CD0D3E             CALL 3E0DH
B96D 23                 INC  HL
B96E 18F7               JR   DSPMES
                  ;
B970 0D0A4472 DRIVE: DEFM 0DH,0AH,'Drive number ? ',0
B974 69766520
B978 6E756D62
B97C 6572203F
B980 2000
B982 43686563 CHKOK: DEFM 'Check Ok !!',0DH,0AH,0
B986 6B204F6B
B98A 2021210D
B98E 0A00
B990 46415420 ERRMES:DEFM 'FAT compare error !!',7,0DH,0AH,0
B994 636F6D70
B998 61726520
B99C 6572726F
B9A0 72202121
B9A4 070D0A00
                  ;
B9A8              FATBFR:DEFS 160
                  ;
BA48                    END


a=&hb900:call a

Drive number ? 1
Check Ok !!

Drive number ? 2
Check Ok !!

Drive number ?
Ok
```

Ａ３２７Ｈ　ディレクトリの読み出し．

アドレス　Ａ３２７Ｈ

機能　ディレクトリをメモリ上に読み出す．

レジスタ　Ａ，Ｆ，Ｂ，Ｃ，Ｈ，Ｌ

解説　ＥＣ８３Ｈ，８４Ｈ両番地のアドレスで示されるインプット・バッファに，Ｄレジスタの値で示されるディレクトリ・セクタを読み出すサブルーチンです．

ディレクトリは複数のセクタにわたって書き込まれており，このサブルーチンではＤレジスタにディレクトリ・セクタの番号（５インチで１～１２，８インチで１～２２）の値を与えることで，どのセクタかを指定します．

リターン時には，ＨＬレジスタにバッファの先頭アドレスが与えられており，またエラーの発生した場合には，Disk Ｉ／Ｏ error が起こります．

サンプル

```
                ;
                ; --- display all file name,type,attribute,top of cluster
                ;
                ;       sample of subroutine from A327H
                ;
                        ORG   0C000H
                ;
                ; getting drive number and set disk ID
                ;
C000 21EBC0     ENT:    LD    HL,DRIVE          ; input drive number
C003 CD6DC0             CALL  DSPMES
C006 CDC85F             CALL  5FC8H             ; subroutine line input
C009 D8                 RET   C                 ; stop key pushed,return to OS
                ;
C00A CDE0C0             CALL  CRLF
C00D 23                 INC   HL
C00E CDBC26             CALL  26BCH             ; convert to binary code
C011 CDA021             CALL  21A0H             ; convert FAC to integer
                ;
C014 3A41EC             LD    A,(0EC41H)        ; get lower byte
C017 CD5399             CALL  9953H             ; set disk identification
                ;
                ; main
                ;
C01A 1601       MAIN:   LD    D,1               ; set top of directory sector
                ;
C01C CD27A3     NEXTSC:CALL 0A327H              ; read directory
                ;
C01F 1E10               LD    E,16              ; number of file a sector
                ;
C021 D5         NEXTFN:PUSH DE
C022 E5                 PUSH  HL
C023 7E                 LD    A,(HL)            ; get top charactor of file name
C024 FEFF               CP    0FFH
C026 2841               JR    Z,EXIT            ; end of directory
C028 B7                 OR    A
C029 282A               JR    Z,NEXT            ; killed file
                ;
C02B CD76C0             CALL  DSPFIL            ; display file name
                ;
```

```
C02E 7E                 LD   A,(HL)             ;'get file type
C02F CD87C0             CALL DSPTYP
               ;
C032 7E                 LD   A,(HL)
C033 E67E               AND  7EH                ; only attribute
               ;
C035 CDD3C0             CALL CLRATR             ; clear attribute save area
               ;
C038 4F                 LD   C,A
C039 CB71               BIT  6,C
C03B C4C1C0             CALL NZ,ROPT            ; read after write option
C03E CB69               BIT  5,C
C040 C4C7C0             CALL NZ,EOPT            ; P option
C043 CB61               BIT  4,C
C045 C4CDC0             CALL NZ,POPT            ; write protect file
               ;
C048 CDADC0             CALL DSPATR             ; dosplay attribute
               ;
C04B 23                 INC  HL
C04C 6E                 LD   L,(HL)             ; get top of cluster
C04D 2600               LD   H,0
C04F CDC228             CALL 28C2H
               ;
C052 CDE0C0             CALL CRLF               ; line feed
               ;
C055 E1        NEXT:    POP  HL
C056 D1                 POP  DE
               ;
C057 D5                 PUSH DE
C058 111000             LD   DE,16
C05B 19                 ADD  HL,DE
C05C D1                 POP  DE
               ;
C05D 1D                 DEC  E
C05E 20C1               JR   NZ,NEXTFN          ; next file name
               ;
C060 14                 INC  D
C061 3AAFEC             LD   A,(0ECAFH)         ; last sector of directory +1
C064 BA                 CP   D
C065 20B5               JR   NZ,NEXTSC          ; next sector
               ;
C067 1897               JR   ENT
               ;
C069 E1        EXIT:    POP  HL
C06A D1                 POP  DE
C06B 1893               JR   ENT
               ;
               ; subroutine display message
               ;
C06D 7E        DSPMES:LD   A,(HL)
C06E A7                 AND  A
C06F C8                 RET  Z
C070 CD0D3E             CALL 3E0DH              ; output 1 byte to CRT
C073 23                 INC  HL
C074 18F7               JR   DSPMES
               ;
               ; subroutine display file name
               ;
C076 F5        DSPFIL:PUSH AF
C077 0609               LD   B,9                ;, length of file name
C079 7E        DSPFLL:LD   A,(HL)
C07A CD0D3E             CALL 3E0DH
C07D 23                 INC  HL
C07E 10F9               DJNZ DSPFLL
C080 3E0E               LD   A,14
C082 3287EF             LD   (0EF87H),A         ; set x axis
C085 F1                 POP  AF
C086 C9                 RET
               ;
               ; subroutine display file type
               ;
C087 E5        DSPTYP:PUSH HL
C088 0600               LD   B,0
C08A 0F                 RRCA
```

```
C08B 3806           JR    C,DSPTPM              ; machine file
C08D 04             INC   B
C08E 07             RLCA
C08F 07             RLCA
C090 3801           JR    C,DSPTPM              ; binary file
C092 04             INC   B
               ;
C093 78        DSPTPM:LD    A,B
C094 87             ADD   A,A
C095 2113C1         LD    HL,TYPTBL             ; table of message of type
C098 0600           LD    B,0
C09A 4F             LD    C,A
C09B 09             ADD   HL,BC
C09C D5             PUSH  DE
C09D 7E             LD    A,(HL)
C09E 23             INC   HL
C09F 56             LD    D,(HL)
C0A0 5F             LD    E,A
C0A1 EB             EX    DE,HL
C0A2 D1             POP   DE
               ;
C0A3 CD6DC0         CALL  DSPMES
C0A6 3E18           LD    A,24
C0A8 3287EF         LD    (0EF87H),A            ; set x axis
               ;
C0AB E1             POP   HL
C0AC C9             RET
               ;
               ; subroutine display attribute
               ;
C0AD E5        DSPATR:PUSH HL
C0AE 2110C1         LD    HL,ATRSAV             ; attribute save area
C0B1 0603           LD    B,3
C0B3 7E        DSPATL:LD    A,(HL)
C0B4 CD0D3E         CALL  3E0DH
C0B7 23             INC   HL
C0B8 10F9           DJNZ  DSPATL
C0BA 3E1E           LD    A,30
C0BC 3287EF         LD    (0EF87H),A            ; set x axis
C0BF E1             POP   HL
C0C0 C9             RET
               ;
               ; subroutine set option to working
               ;
C0C1 3E52      ROPT:  LD    A,'R'
C0C3 3210C1         LD    (ATRSAV),A
C0C6 C9             RET
               ;
C0C7 3E45      EOPT:  LD    A,'E'
C0C9 3211C1         LD    (ATRSAV+1),A
C0CC C9             RET
               ;
C0CD 3E50      POPT:  LD    A,'P'
C0CF 3212C1         LD    (ATRSAV+2),A
C0D2 C9             RET
               ;
               ; subroutine clear attribute save area
               ;
C0D3 E5        CLRATR:PUSH HL
C0D4 2110C1         LD    HL,ATRSAV
C0D7 0603           LD    B,3
C0D9 3620      CLRATL:LD    (HL),' '            ; space
C0DB 23             INC   HL
C0DC 10FB           DJNZ  CLRATL
C0DE E1             POP   HL
C0DF C9             RET
               ;
               ; subroutine line feed
               ;
C0E0 3E0D      CRLF:  LD    A,0DH               ; carriage return
C0E2 CD0D3E         CALL  3E0DH
C0E5 3E0A           LD    A,0AH                 ; line feed
C0E7 CD0D3E         CALL  3E0DH
C0EA C9             RET
```

```
                ;
C0EB 0D0A4452 DRIVE: DEFM 0DH,0AH,'DRIVE NUMBER:',0
C0EF 49564520
C0F3 4E554D42
C0F7 45523A00
                ;
C0FB 4D414348 MACHIN:DEFM 'MACHINE',0
C0FF 494E4500
C103 42494E41 BINARY:DEFM 'BINARY',0
C107 525900
C10A 41534349 ASCII: DEFM 'ASCII';0
C10E 4900
                ;
C110          ATRSAV:DEFS 3
                ;
C113 FBC0     TYPTBL:DEFW MACHIN
C115 03C1            DEFW BINARY
C117 0AC1            DEFW ASCII
                ;
C119                 END
```

```
a=&hc000:call a

DRIVE NUMBER:1

backupn88     BINARY          72
xfilesn88     BINARY      E   73
formatn88     ASCII           69
sysgen        BINARY     R    76
dummy dat     ASCII           66
dummy bin     MACHINE         78
kill  dat     ASCII        P  65

DRIVE NUMBER:
Ok
```

Ａ３４０Ｈ　ディレクトリの書き込みを行う．

アドレス　Ａ３４０Ｈ

機　　能　ディレクトリのディスクへの書き込みを行う．

レジスタ　Ａ，Ｆ，Ｂ，Ｃ，Ｈ，Ｌ

解　　説　ＥＣ８３Ｈ，８４Ｈ両番地のアドレスで示されるインプット・バッファより，Ｄレジスタの値で示されるディレクトリ・セクタへの書き込みを行なうサブルーチンです．

ディレクトリは複数のセクタにわたって書き込まれており，このサブルーチンではＤレジスタにディレクトリ・セクタの番号（５インチで１～１２，８インチで１～２２）の値を与えることで，どのセクタかを指定します．

リターン時には，ＨＬレジスタにバッファの先頭アドレスが与えられており，またエラーの発生した場合には Disk Ｉ／Ｏ error が起こります．

サンプル

```
                ;
                ; --- set file attribute
                ;
                ;       sample of subroutine from A340H
                ;
                        ORG   0B900H
                ;
B900 215BB9     STFATR:LD     HL,FILENM          ; input file name
B903 CD52B9            CALL   DSPMES
B906 CDC85F            CALL   5FC8H              ; line input subroutine
B909 D8                RET    C                  ; stop key return
                ;
B90A 3622              LD     (HL),'"'
B90C CD8C46            CALL   468CH              ; processing file descriptor
B90F A7                AND    A
B910 FA00B9            JP     M,STFATR
                ;
B913 CD5899            CALL   9958H              ; set disk identification
                ;
B916 CDEA98            CALL   98EAH              ; directory search
B919 20E5              JR     NZ,STFATR          ; not found
                ;
B91B 216AB9            LD     HL,ATTRBT          ; input attribute
B91E CD52B9            CALL   DSPMES
B921 CDC85F            CALL   5FC8H
                ;
B924 0E00              LD     C,0
B926 23                INC    HL
B927 7E                LD     A,(HL)             ; get attribute word
B928 FE50              CP     'P'                ; write protect
B92A 2002              JR     NZ,SKIPP
B92C CBE1              SET    4,C                ; set WP byte
                ;
B92E FE52       SKIPP: CP     'R'                ; read after write
B930 2002              JR     NZ,SKIPR
B932 CBF1              SET    6,C                ; set RAF byte
                ;
B934 C5         SKIPR: PUSH   BC
B935 3A8DEC            LD     A,(0EC8DH)         ; get sector of file
B938 57                LD     D,A
```

```
B939 CD27A3          CALL 0A327H          ; directory read
B93C C1              POP  BC
B93D 2A8AEC          LD   HL,(0EC8AH)     ; get address in buffer
B940 110900          LD   DE,9
B943 19              ADD  HL,DE
B944 7E              LD   A,(HL)          ; get attribute
B945 E6AF            AND  0AFH            ; mask except R,P bit
B947 B1              OR   C               ; mix
B948 77              LD   (HL),A
                ;
B949 3A8DEC          LD   A,(0EC8DH)
B94C 57              LD   D,A
B94D CD40A3          CALL 0A340H          ; directory write
                ;
B950 18AE            JR   STFATR
                ;
B952 7E         DSPMES:LD A,(HL)          ; display message subroutine
B953 A7              AND  A
B954 C8              RET  Z
B955 CD0D3E          CALL 3E0DH
```

```
B958 23              INC  HL
B959 18F7            JR   DSPMES
                ;
B95B 0D0A4669 FILENM:DEFM 0DH,0AH,'File name ? ',0
B95F 6C65206E
B963 616D6520
B967 3F2000
                ;
B96A 41747472 ATTRBT:DEFM 'Attribute word ? ',0
B96E 69627574
B972 6520776F
B976 7264203F
B97A 2000
                ;
B97C                 END
```

```
? attr$("backup.n88")

Ok
a=&hb900:call a

File name ? backup.n88
Attribute word ? P

File name ?
Ok
? attr$("backup.n88")
  P
Ok
```

Ａ４ＦＦＨ　クラスタ・トラック変換．

アドレス　Ａ４ＦＦＨ

機　　能　クラスタ・セクタナンバよりトラック・セクタナンバを得る．

レジスタ　Ａ，Ｆ，Ｂ，Ｃ

解　　説　Ｂレジスタにクラスタ・ナンバ，Ｃレジスタにそのクラスタ内におけるセクタ・ナンバを与えてＣＡＬＬすることにより，Ｂレジスタにトラック・ナンバ（論理トラック），Ｃレジスタにそのトラック内におけるセクタ・ナンバを計算して与えます．

このサブルーチンでは，値はワークエリアを参照して計算しているため，このサブルーチンのＣＡＬＬの前には４７４２Ｈあるいは９９５３Ｈ番地のサブルーチンを利用してワークエリアに情報をセットしておく必要があります．

サンプル

```
                ;
                ; --- convert cluster,sector to track,sector
                ;
                ;       sample of subroutine from A4FFH
                ;
                        ORG   0B900H
                ;
B900 2160B9     CLSTRK:LD    HL,DRIVE            ; input drive number
B903 CD3BB9             CALL GETNUM
B906 D8                 RET   C                  ; stop key return
B907 CD5399             CALL 9953H               ; set disk identification
                ;
B90A 2172B9             LD    HL,CLST            ; input cluster number
B90D CD3BB9             CALL GETNUM
B910 47                 LD    B,A
                ;
B911 2184B9             LD    HL,SECTOR          ; input sector number
B914 CD3BB9             CALL GETNUM
B917 4F                 LD    C,A
                ;
B918 CDFFA4             CALL 0A4FFH              ; convert
                ;
B91B 2195B9             LD    HL,TRACK           ; print track
B91E CD57B9             CALL DSPMES
B921 78                 LD    A,B
B922 CD4EB9             CALL DSPACC
                ;
B925 219CB9             LD    HL,DSEC            ; print sector
B928 CD57B9             CALL DSPMES
B92B 79                 LD    A,C
B92C CD4EB9             CALL DSPACC
                ;
B92F 3E0D               LD    A,0DH              ; line feed
B931 CD0D3E             CALL 3E0DH
B934 3E0A               LD    A,0AH
B936 CD0D3E             CALL 3E0DH
                ;
B939 18C5               JR    CLSTRK
                ;
B93B C5         GETNUM:PUSH BC                   ; subroutine get parameter
```

```
B93C CD57B9          CALL DSPMES
B93F CDC85F          CALL 5FC8H         ; line input subroutine
B942 F5              PUSH AF
B943 23              INC  HL
B944 CDBC26          CALL 26BCH         ; convert to binary
B947 CDA021          CALL 21A0H         ; convert to integer
B94A F1              POP  AF
B94B 7D              LD   A,L
B94C C1              POP  BC
B94D C9              RET
                ;
B94E C5         DSPACC:PUSH BC          ; print accumulator
B94F 6F              LD   L,A
B950 2600            LD   H,0
B952 CDC228          CALL 28C2H         ; print integer
B955 C1              POP  BC
B956 C9              RET
                ;
B957 7E         DSPMES:LD   A,(HL)
B958 A7              AND  A
B959 C8              RET  Z
B95A CD0D3E          CALL 3E0DH
B95D 23              INC  HL
B95E 18F7            JR   DSPMES
                ;
B960 0D0A4472 DRIVE: DEFM 0DH,0AH,'Drive number ? ',0
B964 69766520
B968 6E756D62
B96C 6572203F
B970 2000
B972 436C7573 CLST:  DEFM 'Cluster number ? ',0
B976 74657220
B97A 6E756D62
B97E 6572203F
B982 2000
B984 53656374 SECTOR:DEFM 'Sector number ? ',0
B988 6F72206E
B98C 756D6265
B990 72203F20
B994 00
                ;
B995 54524143 TRACK: DEFM 'TRACK:',0
B999 4B3A00
B99C 2C534543 DSEC:  DEFM ',SECTOR:',0
B9A0 544F523A
B9A4 00
                ;
B9A5                 END
```

```
a=&hb900:call a

Drive number ? 1
Cluster number ? 159
Sector number ? 8
TRACK:79,SECTOR:16

Drive number ? 2
Cluster number ? 79
Sector number ? 1
TRACK:39,SECTOR:9

Drive number ?
Ok
```

Ａ５３ＢＨ　ディスクとの入出力を行う．

アドレス	Ａ５３ＢＨ
機　能	ディスクとの複数セクタの入出力を行う．
レジスタ	Ｆ，Ｄ，Ｅ

解　説　ディスクとの入出力を，複数セクタにわたりグラフィックＲＡＭを除く全てのＲＡＭに対して行うサブルーチンです．与えるパラメータは以下の様になり，３６９ＡＨ番地からのサブルーチンに準じます．

ＣＹフラグ　………１：ライト，０：リード
Ｚフラグ　………１：ベリファイ，０：リード
Ａレジスタ　………セクタ数
Ｂレジスタ　………トラック・ナンバ
Ｃレジスタ　………セクタ・ナンバ
ＨＬレジスタ………データ・アドレス
ＥＣ８５Ｈ　………ドライブ・ナンバ（０～）
ＥＣ８６Ｈ，８７Ｈ
　　………ＥＣ８５Ｈ番地の値で示されるドライブのバッファアドレス．
ＥＣＡ７Ｈ　………ドライブが両面の場合１，片面の場合０
ＥＦ５ＤＨ　………ドライブ・タイプ（０～３）

トラックの指定には論理トラック・ナンバを使用するので，サーフェスの指定は必要ありません．また，ドライブがマウント状態であるときには，アクセスしたトラック・ナンバがドライブバッファにセットされます（１－８を参照）．また，２つ以上のトラックをアクセスする様なセクタ数の指定はできません．

エラーの発生した場合には，ＤＩＳＫ Ｉ ／ Ｏ error が起こります．

サンプル

```
                ;
                ; --- set disk information
                ;
                ;       sample of subroutine from A53BH
                ;
                        ORG   0B900H
                ;
B900 2197B9     SETINF:LD    HL,TITLE           ; print title
B903 CD8EB9            CALL  DSPMES
                ;
B906 2A7FEC            LD    HL,(0EC7FH)        ; file buffer address table
B909 5E                LD    E,(HL)
B90A 23                INC   HL
B90B 56                LD    D,(HL)
B90C EB                EX    DE,HL
                ;
B90D 110900            LD    DE,9               ; top of buffer area
B910 19                ADD   HL,DE
                ;
B911 E5                PUSH  HL
                ;
B912 21B2B9     SETINL:LD    HL,INIFN           ; initializing file number
B915 CD8EB9            CALL  DSPMES
B918 CDC85F            CALL  5FC8H              ; line input from keyboard
                ;
B91B 23                INC   HL
B91C 7E                LD    A,(HL)
B91D B7                OR    A                  ; return only
B91E 3EFF              LD    A,0FFH
B920 2811              JR    Z,SETIL1
                ;
B922 CDBC26            CALL  26BCH              ; convert to binary
B925 3ABDEA            LD    A,(0EABDH)         ; check sign of FAC
B928 FE02              CP    2
B92A 20E6              JR    NZ,SETINL
                ;
B92C 3A41EC            LD    A,(0EC41H)         ; lower byte of FAC
B92F FE10              CP    16                 ; maximum file number
B931 30DF              JR    NC,SETINL
                ;
B933 E1         SETIL1:POP   HL
B934 23                INC   HL
B935 77                LD    (HL),A             ; set to buffer
                ;
B936 E5                PUSH  HL
B937 21C3B9            LD    HL,INITXT          ; initializing text
B93A CD8EB9            CALL  DSPMES
B93D CDC85F            CALL  5FC8H
B940 EB                EX    DE,HL
B941 E1                POP   HL
                ;
B942 13         SETIL2:INC   DE
B943 23                INC   HL
B944 1A                LD    A,(DE)
B945 B7                OR    A
B946 2803              JR    Z,SETINE
                ;
B948 77                LD    (HL),A
B949 18F7              JR    SETIL2
                ;
B94B 21E0B9     SETINE:LD    HL,ASKRDY          ; ask write or abort
B94E CD8EB9            CALL  DSPMES
B951 CDC85F            CALL  5FC8H
B954 23                INC   HL
B955 7E                LD    A,(HL)
B956 FE79              CP    'y'                ; yes
B958 20B8              JR    NZ,SETINL
                ;
B95A 3E01              LD    A,1                ; drive# 1
B95C CD5399            CALL  9953H              ; set disk identification
                ;
B95F 3AAAEC            LD    A,(0ECAAH)         ; directory track
```

```
B962 47                 LD   B,A
B963 3AA7EC             LD   A,(0ECA7H)         ; surface flag
B966 B7                 OR   A
B967 78                 LD   A,B
B968 2801               JR   Z,SETIW1
                ;
B96A 87                 ADD  A,A
B96B FE46       SETIW1:CP    70                 ; 8inches
B96D 2802               JR   Z,SETIW2
                ;
B96F C601               ADD  A,1
B971 47         SETIW2:LD    B,A                ; set track number
B972 3AAFEC             LD   A,(0ECAFH)         ; ID sector
B975 4F                 LD   C,A                ; set sector number
                ;
B976 2A7FEC             LD   HL,(0EC7FH)        ; table of file buffer
B979 5E                 LD   E,(HL)
B97A 23                 INC  HL
B97B 56                 LD   D,(HL)
B97C EB                 EX   DE,HL
                ;
B97D 110900             LD   DE,9
B980 19                 ADD  HL,DE
                ;
B981 3E01               LD   A,1                ; number of sector
B983 37                 SCF                     ; write
                ;
B984 CD3BA5             CALL 0A53BH             ; write to disk
                ;
B987 21FBB9             LD   HL,SETMES
B98A CD8EB9             CALL DSPMES
                ;
B98D C9                 RET
                ;
B98E 7E         DSPMES:LD    A,(HL)             ; display subroutine message
B98F A7                 AND  A
B990 C8                 RET  Z
B991 CD0D3E             CALL 3E0DH
B994 23                 INC  HL
B995 18F7               JR   DSPMES
                ;
B997 0D0A5B5B   TITLE: DEFM 0DH,0AH,'[[ Set to ID sector ]]',0DH,0AH,0
B99B 20536574
B99F 20746F20
B9A3 49442073
B9A7 6563746F
B9AB 72205D5D
B9AF 0D0A00
                ;
B9B2 0D0A4669   INIFN: DEFM 0DH,0AH,'File number ? ',0
B9B6 6C65206E
B9BA 756D6265
B9BE 72203F20
B9C2 00
B9C3 54657874   INITXT:DEFM 'Text(within 254 charactors):',0
B9C7 28776974
B9CB 68696E20
B9CF 32353420
B9D3 63686172
B9D7 6163746F
B9DB 7273293A
B9DF 00
                ;
B9E0 0D0A5772   ASKRDY:DEFM 0DH,0AH,'Write to disk(y/else) ? ',0
B9E4 69746520
B9E8 746F2064
B9EC 69736B28
B9F0 792F656C
B9F4 73652920
B9F8 3F2000
                ;
B9FB 0D0A5365   SETMES:DEFM 0DH,0AH,'Set completed. ,0
B9FF 7420636F
BA03 6D706C65
```

```
BA07 7465642E
BA0B 00
                   ;
BA0C                       END
```

```
a=&hb900:call a

[[ Set to ID sector ]]

File number ? 4
Text(within 254 charactors):cls:print "This is system disk !!

Write to disk(y/else) ? y

Set completed.
Ok
mon

h]^d1,0,23,17
Track 23,Surface 00,Sector 17
0000 00 04 63 6C 73 3A 70 72 69 6E 74 20 22 54 68 69     cls:print "Thi
0010 73 20 69 73 20 73 79 73 74 65 6D 20 64 69 73 6B    s is system disk
0020 20 21 21 00 00 00 00 00 00 00 00 00 00 00 00 00     !!
0030 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0040 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0050 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0060 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0070 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0080 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0090 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00A0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00B0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00C0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00D0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00E0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00F0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
h]^b
Ok
```

ＡＢ１ＤＨ　ディスクの空きクラスタ数を求める．

アドレス	Ａ６１ＤＨ
機　能	指定ドライブの空きクラスタ数を求める．
レジスタ	Ａ，Ｆ，Ｂ，Ｃ，Ｄ，Ｅ，Ｈ，Ｌ

解　説　アキュムレータで指定されるドライブの空きクラスタ数を求め，結果を単精度型のフローティングポイント・アキュムレータ（ＦＡＣ）に返します。

サンプル

```
               ;
               ; --- get free clusters
               ;
               ;       sample of subroutine from A61DH
               ;
                      ORG  0B900H
               ;
B900 212EB9    FRCLST:LD   HL,DRIVE            ; input drive number
B903 CD25B9           CALL DSPMES
B906 CDC85F           CALL 5FC8H               ; line input subroutine
B909 D8               RET  C                   ; stop key return
               ;
B90A 23               INC  HL
B90B CDBC26           CALL 26BCH               ; character to binary code
B90E CDA021           CALL 21A0H               ; convert FAC to integer
B911 7D               LD   A,L
B912 F5               PUSH AF
B913 CD1DA6           CALL 0A61DH              ; read free clusters
B916 F1               POP  AF
               ;
B917 CDD028           CALL 28D0H               ; convert to charactor
B91A CD25B9           CALL DSPMES
               ;
B91D 2140B9           LD   HL,ANSWER
B920 CD25B9           CALL DSPMES
               ;
B923 18DB             JR   FRCLST
               ;
               ; subroutine display message
               ;
B925 7E        DSPMES:LD   A,(HL)
B926 A7               AND  A
B927 C8               RET  Z                   ; if end code return
B928 CD0D3E           CALL 3E0DH               ; one character output to CRT
B92B 23               INC  HL
B92C 18F7             JR   DSPMES
               ;
               ; message area
               ;
B92E 0D0A4472  DRIVE: DEFM 0DH,0AH,'Drive number ? ',0
B932 69766520
B936 6E756D62
B93A 6572203F
B93E 2000
B940 20636C75  ANSWER:DEFM ' clusters free.',0
B944 73746572
B948 73206672
B94C 65652E00
               ;
```

```
B950                    END
```

```
a=&hb900:call a

Drive number ? 1
 72 clusters free.
Drive number ? 2
 125 clusters free.
Drive number ?
Ok
```

Ａ６７９Ｈ　ディスクファイルの大きさを求める．

アドレス　Ａ６７９Ｈ

機　　能　ディスクファイルの大きさを求める．

レジスタ　Ａ，Ｆ，Ｂ，Ｃ，Ｄ，Ｅ，Ｈ，Ｌ

解　　説　ＨＬレジスタの値をファイルバッファのアドレスとしたファイルの大きさを求め，ＨＬレジクタ及び整数型ＦＡＣ（フローティングポイント・アキュムレータ）にその値を与えるサブルーチンです．この値は，ＢＡＳＩＣのＬＯＦ関数で返されるものと同一ですが，このサブルーチンはディスクファイルに対してのみ用いることができ，ＲＳ－２３２Ｃに対しては用いることができません．返す値は，シーケンシャルファイルの場合はファイルを構成するセクタ数を，ランダムファイルの場合は最大レコード番号と，.それぞれ見なすことができます．

パラメータとしては，ＨＬレジスタの他にＢレジスタに指定ファイルの存在するデバイス・ナンバ（１－９を参照）を必要とします．また，指定ファイルが開いていなかった場合にはFile not OPENエラーが発生します．

サンプル

```
                 ;
                 ; get large of file
                 ;
                 ;        sample of subroutine from A679H
                 ;
                        ORG   0B900H
                 ;
B900 2143B9      GETLOF:LD    HL,FILENM          ; input file name
B903 CD3AB9             CALL  DSPMES
B906 CDC85F             CALL  5FC8H              ; line input from keyboard
B909 D8                 RET   C                  ; stop key,return
                 ;
B90A 3622               LD    (HL),'"'           ; set double quatation
B90C CD8C46             CALL  468CH              ; processing of file-descriptor
                 ;
B90F 3E01               LD    A,1                ; file#=1
B911 F5                 PUSH  AF
B912 1E04               LD    E,4                ; random file
B914 CDF647             CALL  47F6H              ; file open
B917 F1                 POP   AF
                 ;
B918 F5                 PUSH  AF
B919 CD3547             CALL  4735H              ; set filebuffer address
B91C CD79A6             CALL  0A679H             ; get large of file
B91F F1                 POP   AF
B920 CD1D48             CALL  481DH              ; close file
                 ;
B923 E5                 PUSH  HL                 ; save large of file
B924 2152B9             LD    HL,FILEMS
B927 CD3AB9             CALL  DSPMES
B92A E1                 POP   HL
                 ;
B92B CDC228             CALL  28C2H              ; output integer subroutine
                 ;
B92E 3E0D               LD    A,0DH
B930 CD0D3E             CALL  3E0DH
```

```
B933 3E0A              LD   A,0AH
B935 CD0D3E            CALL 3E0DH
                ;
B938 18C6              JR   GETLOF
                ;
B93A 7E         DSPMES:LD   A,(HL)
B93B A7                AND  A
B93C C8                RET  Z
B93D CD0D3E            CALL 3E0DH
B940 23                INC  HL
B941 18F7              JR   DSPMES
                ;
B943 0D0A4669 FILENM:DEFM 0DH,0AH,'File name ? ',0
B947 6C65206E
B94B 616D6520
B94F 3F2000
                ;
B952 4C617267 FILEMS:DEFM 'Large of file:',0
B956 65206F66
B95A 2066696C
B95E 653A00
                ;
B961                   END
```

```
a=&hb900:call a


File name ? backup.n88
Large of file:14

File name ? setinf.n88
Large of file:5

File name ?
Ok
```

Ａ６Ｆ１Ｈ　ＩＤセクタを読み出す．

アドレス	Ａ６Ｆ１Ｈ
機　　能	ＩＤセクタをメモリ上に読み出す．
レジスタ	Ａ，Ｆ，Ｂ，Ｃ，Ｄ，Ｅ，Ｈ，Ｌ

解　　説　ＩＤセクタを無条件に本体のバッファに読み出すサブルーチンです．リターン時にはＨＬレジスタにバッファの先頭アドレスが格納されています．

エラーの発生の場合，無条件にＡＥＢＡＨ番地にジャンプします．

サンプル

```
              ;
              ; --- get drive attribute and initialize text
              ;
              ;       sample of subroutine from A6F1H
              ;
                     ORG   0B900H
              ;
B900 CDF1A6   RDIDSC:CALL  0A6F1H            ; read ID sector
B903 2A86EC          LD    HL,(0EC86H)       ; get top of buffer
              ;
B906 46              LD    B,(HL)            ; drive attribute
B907 23              INC   HL
B908 4E              LD    C,(HL)            ; maximum file number
B909 E5              PUSH  HL
              ;
B90A 216DB9          LD    HL,ATTR           ; attribute code output
B90D CD52B9          CALL  DSPMES
B910 CB70            BIT   6,B               ; 'R'
B912 3E52            LD    A,'R'
B914 C40D3E          CALL  NZ,3E0DH
B917 CB60            BIT   4,B               ; 'P'
B919 3E50            LD    A,'P'
B91B C40D3E          CALL  NZ,3E0DH
              ;
B91E 3E2E            LD    A,'.'
B920 CD0D3E          CALL  3E0DH
B923 CD5BB9          CALL  LNFEED            ; line feed
              ;
B926 2181B9          LD    HL,MAXFIL         ; maximum file output
B929 CD52B9          CALL  DSPMES
B92C 79              LD    A,C
B92D 3C              INC   A
B92E 281A            JR    Z,NOTDEF          ; FFH
              ;
B930 69              LD    L,C
B931 2600            LD    H,0
B933 CDC228          CALL  28C2H             ; output integer
B936 CD5BB9   RDIDSL:CALL  LNFEED
              ;
B939 2199B9          LD    HL,TXTINT         ; initialize text output
B93C CD52B9          CALL  DSPMES
B93F E1              POP   HL
B940 CD52B9          CALL  DSPMES
B943 CD62B9          CALL  CLRBFR            ; clear input buffer
B946 CD5BB9          CALL  LNFEED
              ;
B949 C9              RET
```

```
                  ;
B94A 21B0B9     NOTDEF:LD    HL,NDMESS
B94D CD52B9            CALL DSPMES
B950 18E4              JR   RDIDSL
                  ;
                  ; subroutine display message
                  ;
B952 7E         DSPMES:LD    A,(HL)
B953 A7                AND  A
B954 C8                RET  Z
B955 CD0D3E            CALL 3E0DH
B958 23                INC  HL
B959 18F7              JR   DSPMES
```

```
                  ;
                  ; subroutine line feed
                  ;
B95B 21BDB9     LNFEED:LD    HL,CRLF
B95E CD52B9            CALL DSPMES
B961 C9                RET
                  ;
                  ; subroutine clear buffer
                  ;
B962 2A86EC     CLRBFR:LD    HL,(0EC86H)
B965 0600              LD   B,0
B967 3600       CLRBF1:LD    (HL),0
B969 23                INC  HL
B96A 10FB              DJNZ CLRBF1
B96C C9                RET
                  ;
                  ; message area
                  ;
B96D 44726976 ATTR:   DEFM 'Drive attribute is ',0
B971 65206174
B975 74726962
B979 75746520
B97D 69732000
B981 4D617869 MAXFIL:DEFM 'Maximum file number is ',0
B985 6D756D20
B989 66696C65
B98D 206E756D
B991 62657220
B995 69732000
B999 496E6974 TXTINT:DEFM 'Initialize text is ...',0
B99D 69616C69
B9A1 7A652074
B9A5 65787420
B9A9 6973202E
B9AD 2E2E00
B9B0 6E6F7420 NDMESS:DEFM 'not defined.',0
B9B4 64656669
B9B8 6E65642E
B9BC 00
B9BD 0D0A00   CRLF:   DEFM 0DH,0AH,0
                  ;
B9C0                   END
```

```
a=&hb900:call a
Drive attribute is R.
Maximum file number is 4
Initialize text is ...run "backup.n88
Ok
```

A74AH マシン語のセーブを行なう.

アドレス	A74AH
機　能	マシン語をファイル上にセーブする.
レジスタ	A, F, B, C, D, E

解　説　メモリ上のデータを直接ファイル上にセーブするサブルーチンで, BSAVE文のメイン処理を行うものです. 与えるパラメータは以下のとおりです.

BCレジスタ………先頭アドレス.
Dレジスタ　………デバイス・ナンバ（1－9を参照).
E653H　………BSAVE文であることを示すフラグ（00H以外).
EC8FH～EC97H
　　　　………ファイル名.
ECB1H, B2H
　　　　………終了アドレス＋1.

また, このサブルーチンにはテキスト解釈も含まれているため, テキストポインタ（HLレジスタ）に00Hをポイントさせる必要があります. それに, リファレンスマニュアルにあるとおりカセットファイルへのセーブは行なうことができません（Bad file name エラーが発生). また, アドレスが不都合な場合には Illegal function call エラーが発生します.

サンプル

```
                ;
                ; --- save machine language program
                ;
                ;        sample of subroutine from A74AH
                ;
                        ORG  0B900H
                ;
B900 2148B9     SAVMLP:LD   HL,FILENM          ; input file name
B903 CD3FB9            CALL DSPMES
B906 CDC85F            CALL 5FC8H              ; line input from keyboard
B909 D8                RET  C                  ; stop key return
                ;
B90A 3622              LD   (HL),'"'
B90C CD8C46            CALL 468CH              ; processing of file descriptor
                ;
B90F E5                PUSH HL
B910 3EFF              LD   A,0FFH             ; set bsave flag
B912 3253E6            LD   (0E653H),A
                ;
B915 2157B9            LD   HL,TOPADR          ; input top address
B918 CD2DB9            CALL GETADR
B91B 44                LD   B,H
```

```
B91C 4D                LD   C,L
                ;
B91D 2166B9            LD   HL,ENDADR          ; input end address
B920 CD2DB9            CALL GETADR
B923 23                INC  HL
B924 22B1EC            LD   (0ECB1H),HL
                ;
B927 E1                POP  HL
B928 CD4AA7            CALL 0A74AH             ; save to file
                ;
B92B 18D3              JR   SAVMLP
                ;
B92D C5         GETADR:PUSH BC                 ; get address subroutine
B92E D5                PUSH DE
B92F CD3FB9            CALL DSPMES
B932 CDC85F            CALL 5FC8H
                ;
B935 23                INC  HL
B936 CDBC26            CALL 26BCH              ; convert to binary
B939 CDA021            CALL 21A0H              ; convert to integer
                ;
B93C D1                POP  DE
B93D C1                POP  BC
B93E C9                RET
                ;
B93F 7E         DSPMES:LD   A,(HL)             ; display message subroutine
B940 A7                AND  A
B941 C8                RET  Z
B942 CD0D3E            CALL 3E0DH
B945 23                INC  HL
B946 18F7              JR   DSPMES
                ;
B948 0D0A4669 FILENM:DEFM 0DH,0AH,'File name ? ',0
B94C 6C65206E
B950 616D6520
B954 3F2000
B957 546F7020 TOPADR:DEFM 'Top address ? ',0
B95B 61646472
B95F 65737320
B963 3F2000
B966 456E6420 ENDADR:DEFM 'End address ? ',0
B96A 61646472
B96E 65737320
B972 3F2000
                ;
B975                   END
```

```
a=&hb900:call a

File name ? machine
Top address ? 0
End address ? &h7ff

File name ?
Ok
```

A7BBH マシン語のロード及び実行を行なう.

アドレス	A7BBH
機　能	マシン語をファイル上からロードし，実行も行なう.
レジスタ	A，F，B，C，D，E，H，L

解　説 ファイル上のマシン語プログラム及びデータをメモリ上にロードするサブルーチンで，BLOAD文のメイン処理を行なうものです．与えるパラメータは以下のとおりです．

BCレジスタ………ロードアドレス．E653Hの値が01H以外の時に有効.

Dレジスタ　………デバイス・ナンバ（1～9を参照).

E653H　………ロードアドレス指定フラグ．01H以外で指定有.

EC8FH～EC97H
………ファイル名.

EF49H　………ロード後，実行するかのフラグ．00H以外で実行.

また，このサブルーチンにはテキスト解釈も含まれているため，テキストポインタ（HLレジスタ）に00Hをポイントさせる必要があります．及び，リファレンスマニュアルにあるとおりカセットファイルよりのロードはできません(Bad file name エラーが発生).また,ロードする領域は，セーブした時に指定したものと同一ですが，E653H番地による指定があると領域の先頭アドレスをBCレジスタペアにとります.

サンプル

```
                ;
                ; --- load machine language program and execute
                ;
                ;       sample of subroutine from A7BBH
                ;
                        ORG  0B900H
                ;
B900 2165B9     LDMLPE:LD    HL,FILENM           ; input file name
B903 CD5CB9             CALL DSPMES
B906 CDC85F             CALL 5FC8H               ; line input subroutine
B909 D8                 RET  C                   ; stop key return
                ;
B90A 3622               LD   (HL),'"'
B90C CD8C46             CALL 468CH               ; processing of file descriptor
                ;
B90F E5                 PUSH HL
B910 D5                 PUSH DE
B911 2174B9             LD   HL,DESADR           ; ask designation of load address
B914 CD5CB9             CALL DSPMES
B917 CDC85F             CALL 5FC8H
```

```
B91A 23                 INC  HL
B91B 7E                 LD   A,(HL)
B91C FE79               CP   'y'
                ;
B91E F5                 PUSH AF
B91F 3E00               LD   A,0
B921 2801               JR   Z,LDMLPJ
B923 3C                 INC  A
B924 3253E6     LDMLPJ:LD    (0E653H),A        ; set designation flag
B927 F1                 POP  AF
                ;
B928 CC4BB9             CALL Z,GETADR
B92B 44                 LD   B,H
B92C 4D                 LD   C,L
                ;
B92D C5                 PUSH BC
B92E 21ABB9             LD   HL,LDGOFL         ; ask 'load and go
B931 CD5CB9             CALL DSPMES
B934 CDC85F             CALL 5FC8H
B937 23                 INC  HL
B938 7E                 LD   A,(HL)
B939 FE79               CP   'y'
B93B 3EFF               LD   A,0FFH
B93D 2801               JR   Z,LDMLPK
B93F 3C                 INC  A
B940 3249EF     LDMLPK:LD    (0EF49H),A        ; set designation flag
                ;
B943 C1                 POP  BC
B944 D1                 POP  DE
B945 E1                 POP  HL
B946 CDBBA7             CALL 0A7BBH            ; load and execute
                ;
B949 18B5               JR   LDMLPE
                ;
B94B 219BB9     GETADR:LD    HL,LDADDR         ; input load address
B94E CD5CB9             CALL DSPMES
B951 CDC85F             CALL 5FC8H
                ;
B954 23                 INC  HL
B955 CDBC26             CALL 26BCH             ; convert to binary
B958 CDA021             CALL 21A0H             ; convert to integer
B95B C9                 RET
                ;
B95C 7E         DSPMES:LD    A,(HL)            ; display message subroutine
B95D A7                 AND  A
B95E C8                 RET  Z
B95F CD0D3E             CALL 3E0DH
B962 23                 INC  HL
B963 18F7               JR   DSPMES
                ;
B965 0D0A4669 FILENM:DEFM 0DH,0AH,'File name ? ',0
B969 6C65206E
B96D 616D6520
B971 3F2000
B974 44657369 DESADR:DEFM 'Designation of load '
B978 676E6174
B97C 696F6E20
B980 6F66206C
B984 6F616420
B988 61646472           DEFM 'address(y/else) ? ',0
B98C 65737328
B990 792F656C
B994 73652920
B998 3F2000
B99B 4C6F6164 LDADDR:DEFM 'Load address ? ',0
B99F 20616464
B9A3 72657373
B9A7 203F2000
B9AB 446F2079 LDGOFL:DEFM 'Do you need R '
B9AF 6F75206E
B9B3 65656420
B9B7 5220
B9B9 6F707469           DEFM 'option(y/else) ? ',0
B9BD 6F6E2879
```

```
B9C1 2F656C73
B9C5 6529203F
B9C9 2000
                    ;
B9CB                          END
```

```
a=&hb900:call a

File name ? sample.bin
Designation of load address(y/else) ? y
Load address ? &hc000
Do you need R option(y/else) ? y

This program is for the sample !!

File name ?
Ok
```

ＡＤ４８Ｈ　プログラムプロテクトのチェックを行う.

アドレス　ＡＤ４８Ｈ

機　能　プログラムプロテクトのチェックを行う.

レジスタ　Ａ，Ｆ，Ｈ，Ｌ

解　説　現在メモリ上にあるＢＡＳＩＣプログラムがＰオプションセーブされているものならば，現在実行中の行番号を調べダイレクトモードであればIllegal function callエラーを発生させます．プログラム実行中や，プログラムがＰオプション付きでないときはＺフラグをクリアして戻ります.

なお，Ｐオプション付きのチェックは，ＥＣ２９Ｈ番地の値を調べることにより行います（ＥＣ２９Ｈ＝００ＨでＰオプション付き).

サンプル

```
                ;
                ; --- create new command IRESET for protected program
                ;
                ;       sample of subroutine from AD48H
                ;
                        ORG   0F320H
                ;
F320 2133F3     IRESET:LD     HL,IRESM          ; jump table set
F323 22AEEE             LD    (0EEADH+1),HL
                ;
F326 210000             LD    HL,0
F329 2288EC             LD    (0EC88H),HL
F32C 213DF3             LD    HL,IREMES
F32F CD5055             CALL  5550H
                ;
F332 C9                 RET
                ;
F333 2B         IRESM: DEC    HL
F334 D7                 RST   10H               ; check syntax
F335 C29303             JP    NZ,393H           ; Syntax error
                ;
F338 AF                 XOR   A                 ; clear P option flag
F339 3229EC             LD    (0EC29H),A
                ;
F33C C9                 RET
                ;
F33D 0D0A4372   IREMES:DEFM 0DH,0AH,'Create IRESET command !!',0
F341 65617465
F345 20495245
F349 53455420
F34D 636F6D6D
F351 616E6420
F355 212100
                ;
F358                    END
```

```
a=&hf320:call a

Create IRESET command !!
Ok
load "test
```

```
Ok
list
Illegal function call
Ok
ireset
Ok
list
10 '
20 ' test program
30 '
40 WIDTH 80,25:CONSOLE 0,25
50 '
60 PRINT "This is a test program for P option !!
70 '
80 END
Ok
```

ＡＤ５ＢＨ　ライトプロテクトのチェックを行う.

アドレス　ＡＤ５ＢＨ

機　　能　ディスクがライトプロテクト状態にあるか調べる.

レジスタ　Ｈ，Ｌ

解　　説　ＥＣ８５Ｈにドライブ・ナンバ（０～）をセットしてコールすることにより，指定ドライブのライトプロテクトの状態をチェックします．指定ドライブがライトプロテクト状態にあるときは，File write protected エラが発生し，それ以外のときは，ＨＬレジスタに指定ドライブのドライブバッファのアドレスをセットし戻ります.

サンプル

```
                     ;
                     ; --- create check Write Protect command STATUS
                     ;
                     ;        sample of subroutine from AD5BH
                     ;
                            ORG   0F320H
                     ;
F320 2133F3          STATUS:LD    HL,STATM          ; jump table set
F323 22B1EE                 LD    (0EEB0H+1),HL
                     ;
F326 210000                 LD    HL,0
F329 2288EC                 LD    (0EC88H),HL
F32C 2151F3                 LD    HL,STTMES
F32F CD5055                 CALL  5550H
                     ;
F332 C9                     RET
                     ;
F333 1E01            STATM: LD    E,1
F335 2808                   JR    Z,STATMJ
                     ;
F337 CDA318                 CALL  18A3H             ; get drive number
F33A 2B                     DEC   HL
F33B D7                     RST   10H               ; check syntax
F33C C29303                 JP    NZ,393H
                     ;
F33F E5              STATMJ:PUSH  HL
F340 3A7DEC                 LD    A,(0EC7DH)        ; maximum drive number
F343 BB                     CP    E
F344 DA13A7                 JP    C,0A713H          ; Bad drive number
F347 7B                     LD    A,E
F348 3D                     DEC   A
F349 3285EC                 LD    (0EC85H),A
                     ;
F34C CD5BAD                 CALL  0AD5BH            ; check WP
                     ;
F34F E1                     POP   HL
F350 C9                     RET
                     ;
F351 0D0A4372        STTMES:DEFM  0DH,0AH,'Create STATUS command !!',0
F355 65617465
F359 20535441
F35D 54555320
F361 636F6D6D
F365 616E6420
F369 212100
                     ;
F36C                        END
```

```
a=&hf320:call a


Create STATUS command !!
Ok
status 1
File write protected
Ok
status 2
Ok
```

第4章　システム・ワーキングエリア

——*System Working area*——

４－１　システム・ワーキングエリア．

Ｅ６００Ｈ－Ｅ６０ＤＨ

単精度減算ルーチン用サブルーチン．

Ｅ６０ＥＨ－Ｅ６３４Ｈ

ＲＮＤ関数用ワークエリア．

Ｅ６３５Ｈ－Ｅ６４８Ｈ

ＵＳＲ関数用ジャンプテーブル．未使用状態で０Ｂ０６Ｈ（Illegal function call）のアドレスが入っている．２バイト×10組．

Ｅ６４９Ｈ　ＥＲＲ関数用．直前に発生したエラーのエラーコードが入っている．

Ｅ６４ＡＨ　未使用．

Ｅ６４ＢＨ－Ｅ６５２Ｈ

出力（プリンタ，ＣＲＴ）制御用ワークエリア．

Ｅ６５３Ｈ　ＢＬＯＡＤ，ＢＳＡＶＥ文用フラグ．Ａ８１０Ｈからのサブルーチンで使用（００Ｈで第一パラメータのみ指定あり，０１Ｈで第一・第二パラメータとも指定なし，ＦＦＨで第一・第二パラメータ両方の指定あり）．

Ｅ６５４Ｈ，５５Ｈ

フリーエリアの上限．ＣＬＥＡＲ文の第二パラメータで宣言．

Ｅ６５６Ｈ，５７Ｈ

現在実行中の行番号．ダイレクトモード時でＦＦＦＦＨ．

Ｅ６５８Ｈ，５９Ｈ

テキストエリアの開始アドレス．

Ｅ６５ＡＨ，５ＢＨ

／０，ＯＶエラー用メッセージアドレス．

Ｅ６５ＣＨ－Ｅ６６８Ｈ

ターミナルモードにおいて，ＥＳＣ！で出力するＩＤコード．

Ｅ６６９Ｈ－Ｅ６８３Ｈ

モニタ用ジャンプテーブル，順に３バイトずつ使用．

　Ｇコマンド用ブレークポイント・ジャンプテーブル

　イニシァライズ拡張用（未使用）

　Ｍコマンド用

　^D，^R，^W用

Ｅコマンド拡張用

＾Ａ（ＨＥＬＰ）コマンド拡張用

汎用（未使用）

ＥＥ６８４Ｈ－Ｅ６９５Ｈ

拡張ＲＯＭチェック用ルーチン．

Ｅ６９６Ｈ－Ｅ６９ＡＨ

メインＲＯＭに戻すルーチン．

Ｅ６９ＢＨ　未使用（Ｃ９Ｈのみ）．

Ｅ６９ＣＨ，９ＤＨ

ＲＳＴ１０Ｈで使用．

Ｅ６９ＥＨ　直前にアクセスしたドライブのタイプ．

Ｅ６９ＦＨ－Ｅ６Ａ２Ｈ

ターミナルモード用ワークエリア．

Ｅ６Ａ３Ｈ　ＤＡＴＡ文で使用．００ＨでOut of data.

Ｅ６Ａ４Ｈ　ＣＭＴ・ヘッダサーチフラグ．

Ｅ６Ａ５Ｈ　ターミナルモード用ワークエリア．

Ｅ６Ａ６Ｈ　専用高解像度ディスプレイモードフラグ．

Ｅ６Ａ７Ｈ　カーソル表示フラグ．００Ｈで消去．

Ｅ６Ａ８Ｈ　カーソル表示用コマンド．ＣＲＴＣに送る．

Ｅ６Ａ４Ｈ　ＣＭＴ使用中フラグ．

Ｅ６ＡＡＨ　未使用．

Ｅ６ＡＢＨ，ＡＣＨ

ピリオドで参照される行番号．

Ｅ６ＡＤＨ　インサートモードフラグ．

Ｅ６ＡＥＨ－Ｅ６ＢＡＨ

テキスト画面用ワークエリア．

Ｅ６ＢＢＨ－Ｂ６ＢＥＨ

ＥＤＩＴ用ワークエリア．

Ｅ６ＢＦＨ　ＵＳＡＲＴ８２５１エラー発生フラグ．

Ｅ６Ｃ０Ｈ　出力ポート３０Ｈに出力したデータ．

Ｅ６Ｃ１Ｈ　出力ポート４０Ｈに出力したデータ．

Ｅ６Ｃ２Ｈ　出力ポート３１Ｈに出力したデータ．

Ｅ６Ｃ３Ｈ　出力ポートＥ６Ｈに出力したデータ．

Ｅ６Ｃ４Ｈ，Ｃ５Ｈ

ＶＲＡＭトップアドレス．

Ｅ６Ｃ６Ｈ，Ｃ７Ｈ

タイマ割り込み処理用ワークエリア．

Ｅ６Ｃ８Ｈ　未使用．

Ｅ６Ｃ９Ｈ　Line buffer overflow フラグ．

Ｅ６ＣＡＨ－Ｅ６Ｅ７Ｈ

キー入力用ワークエリア．

Ｅ６Ｅ８Ｈ　ＵＳＡＲＴ８２５１フラグ．

Ｅ６Ｅ９Ｈ，ＥＡＨ

ＣＯＰＹ用ドット出力カウンタ．

Ｅ６ＥＢＨ　ＯＰＥＮ”ＬＰＴ１：”用フラグ．ＦＦＨでＯＰＥＮ．

Ｅ６ＥＣＨ　ＯＰＥＮ”ＣＯＭ：”用ＷＩＤＴＨ．

Ｅ６ＥＤＨ　ＵＳＡＲＴ８２５１用フラグ．

Ｅ６ＥＥＨ　ＯＮ　ＩＮＴＥＲＲＵＰＴ総数．

Ｅ６ＥＦＨ，Ｆ０Ｈ

ＯＮ　ＩＮＴＥＲＲＵＰＴアドレステーブル．

Ｅ６Ｆ１Ｈ　ＯＮ　ＩＮＴＥＲＲＵＰＴフラグ．

Ｅ６Ｆ２Ｈ－Ｅ７９１Ｈ

プログラマブルファンクションキーのデータ．

Ｆ１～Ｆ１０の順に１６バイト×10組．セパレータは００Ｈ．

Ｅ７９２Ｈ－Ｅ７Ｅ１Ｈ

ターミナルモード・ファンクションキーのデータ．

Ｆ６～Ｆ１０の順に１６バイト×５組．セパレータは００Ｈ．

Ｅ７Ｅ２Ｈ－Ｅ７Ｅ７Ｈ

ＬＩＮＥ文用ジャンプテーブル．３バイト×２組．

Ｅ７Ｅ８Ｈ，Ｅ９Ｈ

ＣＬＥＡＲ文第二パラメータで宣言できる上限（通常Ｅ５ＦＦＨ）．

Ｅ７ＥＡＨ－Ｅ８２５Ｈ

各種インタラプト処理ルーチン．

ＲＳ－２３２Ｃ………Ｅ７２ＡＨ
ＶＲＴＣ　　………Ｅ８０８Ｈ
ＣＬＯＣＫ　………Ｅ８０ＥＨ
ＵＳＥＲ　　………Ｅ８１４Ｈ
ＦＤＣ１　　………Ｅ８１ＡＨ

ＦＤＣ２　　………Ｅ８２０Ｈ

Ｅ８２６Ｈ－Ｅ８４ＦＨ

ＭＯＮコマンド処理ルーチン．

Ｅ８５０Ｈ－Ｅ８７８Ｈ

変数ルーチン用ワークエリア．

Ｅ８７９Ｈ－Ｅ９Ｂ８Ｈ

中間言語バッファ．

Ｅ９Ｂ９Ｈ－ＥＡＢ９Ｈ

インプット・バッファ．

ＥＡＢＡＨ　未使用．

ＥＡＢＢＨ　ＣＨＡＩＮ文用．

ＥＡＢＣＨ　ＤＩＭ文用フラグ．

ＥＡＢＤＨ　ＦＡＣ（フローティングポイント・アキュムレータ）の型．（２：整数型，３：文字型，４：単精度型実数，８：倍精度型実数）

ＥＡＢＥＨ，ＢＦＨ

中間言語・リスト変換フラグ．

ＥＡＣ０Ｈ－ＥＡＣＢＨ

ＲＳＴ　１０Ｈ用ワークエリア．

ＥＡＣＣＨ－ＥＡＦ０Ｈ

ストリング用ワークエリア．

ＥＡＦ１Ｈ，Ｆ２Ｈ

フリーエリア上限（スタック下限）．

ＥＡＦ３Ｈ，Ｆ４Ｈ

式評価ルーチン用．

ＥＡＦ５Ｈ，Ｆ６Ｈ

ガーベッジ・コレクション用．

ＥＡＦ７Ｈ，Ｆ８Ｈ

ＦＯＲ文用．

ＥＡＦ９Ｈ，ＦＡＨ

ＤＡＴＡ文用エラー行番号．

ＥＡＦＢＨ－ＥＡＦＥＨ

変数認識用ワークエリア．

ＥＡＦＦＨ－ＥＢ１ＡＨ

テキスト編集・解釈用ワークエリア．

ＥＢ１６Ｈ，１７Ｈ
　ラベル登録エリア・トップアドレス．

ＥＢ１８Ｈ，１９Ｈ
　テキストエンド・アドレス

ＥＢ１ＡＨ　ラベル登録済フラグ．

ＥＢ１ＢＨ，１ＣＨ
　単純変数領域トップアドレス．

ＥＢ１ＤＨ，１ＥＨ
　配列変数領域トップアドレス．

ＥＡ１ＦＨ，２０Ｈ
　フリーエリア・トップアドレス．

ＥＢ２１Ｈ，２２Ｈ
　ＲＥＡＤ文用データポインタ．

ＥＢ２３Ｈ－ＥＢ３ＣＨ
　変数の暗黙型．Ａ～Ｚの順に２６バイト．

ＥＢ３ＤＨ－ＥＣ１４Ｈ
　ＦＮ用ワークエリア．

ＥＣ１５Ｈ　ＩＮＰＵＴ文用カウンタ．

ＥＣ１６Ｈ－ＥＣ１ＥＨ
　ＦＯＲ－ＮＥＸＴ文用ワークエリア．

ＥＣ１ＦＨ　ＯＰＴＩＯＮ　ＢＡＳＥ文で指定した値．

ＥＣ２０Ｈ　ＯＰＴＩＯＮ　ＢＡＳＥ文実行済フラグ．

ＥＣ２１Ｈ　中間言語・リスト変換用．

ＥＣ２２Ｈ　ＩＮＰＵＴ文で’？’を出力するかどうかのフラグ

ＥＣ２３Ｈ，２４Ｈ
　ＣＨＡＩＮ文用．ストリングエリア・トップアドレスの退避用．

ＥＣ２５Ｈ，２６Ｈ
　未使用．

ＥＣ２７Ｈ　ＳＡＶＥ文用，’Ｐ’オプション・フラグ．

ＥＣ２８Ｈ　ＬＯＡＤ文用，’Ｐ’オプション・フラグ．

ＥＣ２９Ｈ　プログラム・プロテクトフラグ．００Ｈ以外でＬＩＳＴ，ＬＬＩＳＴ，ＳＡＶＥ，ＭＯＮ，ＰＥＥＫ，ＰＯＫＥ，ＵＳＲ，ＣＡＬＬの実行がエラーとなる．

ＥＣ２ＡＨ　ＣＨＡＩＮ文用ＭＥＲＧＥフラグ．

ＥＣ２ＢＨ　ＣＨＡＩＮ文用ＤＥＬＥＴＥフラグ．

ＥＣ２ＣＨ，２ＤＨ

　　ＣＨＡＩＮ文用ＤＥＬＥＴＥエンドアドレス．

ＥＣ２ＥＨ，２ＦＨ

　　ＣＨＡＩＮ文用ＤＥＬＥＴＥトップアドレス．

ＥＣ３０Ｈ　ＣＨＡＩＮ文用フラグ．

ＥＣ３１Ｈ，３２Ｈ

　　ＣＨＡＩＮ文用実行行番号．

ＥＣ３３Ｈ－ＥＣ３ＡＨ

　　ＳＷＡＰ文用変数退避ＦＡＣ．

ＥＣ３ＢＨ　ＴＲＯＮ文用フラグ．

ＥＣ３ＣＨ　フローティングポイント・アキュムレータ（ＦＡＣ）の無効桁．

ＥＣ３ＤＨ－ＥＣ４４Ｈ

　　フローティングポイント・アキュムレータ（ＦＡＣ）

　　整数型：ＥＡＢＤＨ＝０２Ｈ

　　　　ＥＣ４１Ｈ　下位８ビット

　　　　ＥＣ４２Ｈ　上位８ビット

　　単精度型実数：

　　　　ＥＡＢＤＨ＝０４Ｈ

　　　　ＥＣ４１Ｈ　仮数部下位８ビット

　　　　ＥＣ４２Ｈ　仮数部中位８ビット

　　　　ＥＣ４３Ｈ　仮数部上位８ビット

　　　　ＥＣ４４Ｈ　指数部８ビット

　　倍精度型実数：

　　　　ＥＡＢＤＨ＝０８Ｈ

　　　　ＥＣ３ＤＨ　仮数部最下位８ビット

　　　　ＥＣ３ＥＨ　仮数部中位８ビット

　　　　ＥＣ３ＦＨ　仮数部中位８ビット

　　　　ＥＣ４０Ｈ　仮数部中位８ビット

　　　　ＥＣ４１Ｈ　仮数部中位８ビット

　　　　ＥＣ４２Ｈ　仮数部中位８ビット

　　　　ＥＣ４３Ｈ　仮数部最上位８ビット

　　　　ＥＣ４４Ｈ　指数部８ビット

　　文字型：ＥＡＢＤＨ＝０３Ｈ

ＥＣ４１Ｈ　ストリング・ディスクリプタ下位
ＥＣ４２Ｈ　ストリング・ディスクリプタ上位

ＥＣ４５Ｈ　フローティングポイント・アキュムレータ（ＦＡＣ）の符号.

ＥＣ４６Ｈ　／０，ＯＶエラーフラグ.

ＥＣ４７Ｈ　ＯＶ出力フラグ.

ＥＣ４８Ｈ　１０進化フラグ.

ＥＣ４９Ｈ　フローティングポイント・レジスタ（ＦＲＥＧ）の無効桁.

ＥＣ４ＡＨ－ＥＣ５１Ｈ
フローティングポイント・レジスタ（ＦＲＥＧ）.

ＥＣ５２Ｈ－ＥＣ６ＣＨ
数値・文字列変換バッファ.

ＥＣ６ＤＨ－ＥＣ７４Ｈ
倍精度除算用レジスタ.

ＥＣ７５Ｈ－ＥＣ７ＣＨ
倍精度乗算用レジスタ.

ＥＣ７ＤＨ　接続されているドライブ数.

ＥＣ７ＥＨ　開くことのできるファイル数.

ＥＣ７ＦＨ，８０Ｈ
ファイルバッファ・アドレステーブルのトップアドレス.

ＥＣ８１Ｈ，８２Ｈ
ドライブバッファ・アドレステーブルのトップアドレス.

ＥＣ８３Ｈ，８４Ｈ
インプットバッファのアドレス.

ＥＣ８５Ｈ　ドライブ・ナンバ.

ＥＣ８６Ｈ，８７Ｈ
ドライブバッファのアドレス.

ＥＣ８８Ｈ，８９Ｈ
ファイルバッファのアドレス.

ＥＣ８ＡＨ，８ＢＨ
ディレクトリサーチ中のファイル名のバッファ内のアドレス.

ＥＣ８ＣＨ　ディレクトリサーチ中のファイル名のセクタ内の番号.

ＥＣ８ＤＨ　ディレクトリサーチ中のファイル名のあるセクタ番号.

ＥＣ８ＥＨ　ＬＯＡＤ文の'Ｒ'オプション・フラグ.

ＥＣ８ＦＨ－ＥＣ９７Ｈ

ファイル名バッファ（I）．一般にこちらのバッファを使用．

ＥＣ９８Ｈ－ＥＣＡ０Ｈ

ファイル名バッファ（II）．ＮＡＭＥ文で使用．

ＥＣＡ１Ｈ　ＤＳＫＩ＄，ＤＳＫＯ＄用のトラック・ナンバ．

ＥＣＡ２Ｈ　ＤＳＫＩ＄，ＤＳＫＯ＄用のセクタ・ナンバ．

ＥＣＡ３Ｈ　ファイルＣＬＯＳＥ用フラグ．

ＥＣＡ４Ｈ　ＳＡＶＥ文用フラグ．

ＥＣＡ５Ｈ　最大トラック・ナンバ（物理トラック）．

ＥＣＡ６Ｈ　１トラックあたりのセクタ数．

ＥＣＡ７Ｈ　サーフェス・フラグ（００Ｈ：両面，０１Ｈ：片面）．

ＥＣＡ８Ｈ　１トラックあたりのクラスタ数．

ＥＣＡ９Ｈ　１ボリュームあたりのクラスタ数．

ＥＣＡＡＨ　ディレクトリのあるトラック番号．

ＥＣＡＢＨ　１クラスタあたりのセクタ数．

ＥＣＡＣＨ　ＦＡＴの開始セクタ番号．

ＥＣＡＤＨ　ＦＡＴの終了セクタ番号．

ＥＣＡＥＨ　ＦＡＴの数．

ＥＣＡＦＨ　ＩＤセクタ番号．

ＥＣＢ０Ｈ　未使用．

ＥＣＢ１Ｈ，Ｂ２Ｈ

ＢＳＡＶＥ文用の終了アドレス．

ＥＣＢ３Ｈ　未使用．

ＥＣＢ４Ｈ　ディスク入出力用エラーカウンタ（I）．

ＥＣＢ５Ｈ　ディスク入出力用エラーカウンタ（II）．

ＥＣＢ６Ｈ　ディスク入出力用エラーフラグ．

ＥＣＢ７Ｈ　ディスク入出力用．

ＥＣＢ８Ｈ　ディスク入出力用．

ＥＣＢ９Ｈ，ＢＡＨ

未使用．

ＥＣＢＢＨ－ＥＥ０ＤＨ

N_{88}－DISK BASIC 用フックアドレステーブル．

４－２を参照．

ＥＥ０ＥＨ－ＥＥ７０Ｈ

N_{88}－ＤＩＳＫ ＢＡＳＩＣ用入出力ルーチンジャンプテーブル．

４－３を参照．

ＥＥ７１Ｈ－ＥＥＣＡＨ
　N_{88}－ＤＩＳＫ ＢＡＳＩＣ用コマンド・ステートメント・関数ジャンプテーブル．６－１を参照．

ＥＥＣＢＨ－ＥＦ０９Ｈ
　ＯＮ　ＩＮＴＥＲＲＵＰＴ　ＧＯＳＵＢ文用アドレステーブル．

ＥＦ０ＡＨ　サブＲＯＭコール時のアキュムレータの値．

ＥＦ０ＢＨ，０ＣＨ
　サブＲＯＭコール時のＨＬレジスタの値．

ＥＦ０ＤＨ　ＣＯＰＹ文用．

ＥＦ０ＥＨ　出力ポート　Ｅ４Ｈへの出力データ．

ＥＦ０ＦＨ　出力ポート　Ｆ４ＨあるいはＦ８Ｈへの出力データ．

ＥＦ１０Ｈ　ディスク用タイムアウトカウンタ．

ＥＦ１１Ｈ　ディスク用イニシァライズ・フラグ．

ＥＦ１２Ｈ，１３Ｈ
　ＤＭＡフロッピ用転送アドレス．

ＥＦ１４Ｈ　ディスク用待時間．

ＥＦ１５Ｈ　ミニフロッピ用の高速転送用フラグ．

ＥＦ１６Ｈ　ＤＭＡタイプ・アクセス中フラグ．

ＥＦ１７Ｈ　ＤＭＡタイプ・コマンド（００Ｈ：イニシァライズ，０２Ｈ：ライト，０３Ｈ：リード）．

ＥＦ１８Ｈ　ＤＭＡタイプ・ドライブ・サーフェス（ビット２：サーフェス，ビット１，０：ドライブ）．

ＥＦ１９Ｈ　ＤＭＡタイプ・トラック・ナンバ．

ＥＦ１ＡＨ　ＤＭＡタイプ・セクタ・ナンバ．

ＥＦ１ＢＨ　ＤＭＡタイプ・転送セクタ数．

ＥＦ１ＣＨ，１ＤＨ
　ＤＭＡタイプ・転送アドレス．

ＥＦ１ＥＨ－ＥＦ２０Ｈ
　ＤＭＡタイプ・６００進タイムカウンタ．

ＥＦ２１Ｈ　ＤＭＡコントローラへのコマンド・バイト（４０Ｈ：リード，８０Ｈ：ライト）．

ＥＦ２２Ｈ　ＦＤＣへのコマンド・バイト（４５Ｈ：ライト，４６Ｈ：リード）．

ＥＦ２３Ｈ　ＦＤＣのステータス・バイト（８６Ｈ：ライト，８７Ｈ：リード）．

ＥＦ２４Ｈ　DMAタイプ・リトライカウンタ.
ＥＦ２５Ｈ　未使用.
ＥＦ２６Ｈ　ＦＤＣ・レザルト・ステータス（ＳＴ０).
ＥＦ２７Ｈ　ＦＤＣ・レザルト・ステータス（ＳＴ１).
ＥＦ２８Ｈ　ＦＤＣ・レザルト・ステータス（ＳＴ２).
ＥＦ２９Ｈ　ＦＤＣ・レザルト・ステータス（Ｃ：シリンダ・ナンバ).
ＥＦ２ＡＨ　ＦＤＣ・レザルト・ステータス（Ｈ：ヘッド・ナンバ).
ＥＦ２ＢＨ　ＦＤＣ・レザルト・ステータス（Ｒ：セクタ・ナンバ).
ＥＦ２ＣＨ　ＦＤＣ・レザルト・ステータス（Ｎ：１セクタの長さ).
ＥＦ２ＤＨ－ＥＦ３４Ｈ
　DMAタイプ・ドライブ・ステータス・テーブル（５インチ).
ＥＦ３５Ｈ－ＥＦ３ＣＨ
　DMAタイプ・ドライブ・ステータス・テーブル（８インチ).
ＥＦ３ＤＨ　DMAタイプ・エラーステータス.
ＥＦ３ＥＨ　DMAタイプ・インタプラスト待ちフラグ.
ＥＦ３ＦＨ　インタラプトレベル退避用.
ＥＦ４０Ｈ　ＦＤＣ・レザルト・ステータス（ＳＴ３).
ＥＦ４１Ｈ－ＥＦ４８Ｈ
　マージンデータ・テーブル．２バイト×４組.
ＥＦ４９Ｈ　ＢＬＯＡＤ文用のＲオプション・フラグ.
ＥＦ４ＡＨ　ディスクの入出力セクタ数.
ＥＦ４ＢＨ，４ＣＨ
　ドライブ・ステータステーブルのアドレス.
　（５インチ：ＥＦ２ＤＨ，８インチ：ＥＦ３５Ｈ)
ＥＦ４ＤＨ　マージンコントロールポートのアドレス.
　（５インチ：Ｆ９Ｈ，８インチ：Ｆ５Ｈ)
ＥＦ４ＥＨ　インタフェース・チェックポートのアドレス.
　（５インチ：Ｆ８Ｈ，８インチ：Ｆ４Ｈ）bit 0 = 0で接続.
ＥＦ４ＦＨ　ＦＤＣコントロール・アドレス.
　（５インチ：Ｆ８Ｈ，８インチ：Ｆ４Ｈ)
ＥＦ５０Ｈ　片面あたりのトラック数（５インチ：２７Ｈ，８インチ，４ＣＨ).
ＥＦ５１Ｈ　センタトラック・ナンバ（５インチ：１４Ｈ，８インチ：２６Ｈ).
ＥＦ５２Ｈ　DMAタイプ・１トラックあたりのセクタ数.
　（５インチ：１０Ｈ，８インチ：１ＡＨ).

ＥＦ５３Ｈ　ＧＡＰ３の長さ（５インチ：３３Ｈ，８インチ：３６Ｈ）．
ＥＦ５４Ｈ　ＦＤＣステータス・ポートアドレス．
（５インチ：ＦＡＨ，８インチ：Ｆ６Ｈ）．
ＥＦ５５Ｈ　ＦＤＣデータ・ポートアドレス．
（５インチ：ＦＢＨ，８インチ：Ｆ７Ｈ）．
ＥＦ５６Ｈ　Ｆ３Ｈに出力するＦＤＣデータ．
（５インチ：１０Ｈ，８インチ：２０Ｈ）．
ＥＦ５７Ｈ　ステップ・レート（５インチ：３ＡＨ，８インチ：ＢＡＨ）．
ＥＦ５８Ｈ　ヘッド・ロード（５インチ：１８Ｈ，８インチ：４２Ｈ）．
ＥＦ５９Ｈ　ＤＭＡＣに送るコマンド（５インチ：Ａ５Ｈ，８インチ：Ａ６Ｈ）．
ＥＦ５ＡＨ　ＤＭＡＣのチャンネル・アドレス．
（５インチ：６０Ｈ，８インチ：６２Ｈ）．
ＥＦ５ＢＨ　ＤＭＡＣのベース・アドレス（５インチ，８インチ：６０Ｈ）．
ＥＦ５ＣＨ　Ｆ３Ｈに出力する値（５インチ：０１Ｈ，８インチ：０２Ｈ）．
ＥＦ５ＤＨ　ドライブ・タイプ（００Ｈ：８インチＤＭＡ，０１Ｈ：５インチＤＭＡ，０２Ｈ：５インチ片面，０３Ｈ：５インチ両面）．
ＥＦ５ＥＨ　未使用．
ＥＦ５ＦＨ　各ドライブタイプごとのドライブ・ナンバ．
ＥＦ６０Ｈ　ＤＭＡタイプ・８インチの総ドライブ数．
ＥＦ６１Ｈ　ＤＭＡタイプ・５インチの総ドライブ数．
ＥＦ６２Ｈ　インテリジェントタイプの総ドライブ数．
ＥＦ６３Ｈ　インテリジェントタイプ・サーフェス数フラグ．
（片面：００Ｈ，両面：８０Ｈ）
ＥＦ６４Ｈ－ＥＦ６ＦＨ
ドライブ・ナンバ，ドライブ・タイプ対応表．
ＥＦ７０Ｈ　未使用．
ＥＦ７１Ｈ－ＥＦ７ＥＨ
ターミナルモード用ワークエリア．
ＥＦ７ＦＨ　入力ポート３０Ｈからのデータ．コンプリメント値．
ＥＦ８０Ｈ　入力ポート３１Ｈからのデータ．コンプリメント値．
ＥＦ８１Ｈ－ＥＦＢ４Ｈ
テキスト画面用ワークエリア．
ＥＦＢ５Ｈ－ＥＦＣ３Ｈ
テキスト解釈用ワークエリア．

ＥＦＣ４Ｈ－ＥＦＣＣＨ

ＩＮＰＵＴ　ＷＡＩＴ文・ＯＮ　ＴＩＭＥ＄　ＧＯＳＵＢ文用カウンタ.

ＥＦＣＤＨ－ＥＦＤ２Ｈ

キー入力用キュー用ワークエリア.

ＥＦＤ３Ｈ－ＥＦＤ８Ｈ

ＵＳＡＲＴ入力用キュー用ワークエリア.

ＥＦＤ９Ｈ－ＥＦＦ８Ｈ

キー入力用キュー.

ＥＦＦ９Ｈ－Ｆ００４Ｈ

キースキャン用ワークエリア.

Ｆ００５Ｈ　ＵＳＡＲＴのファイル属性.

Ｆ００６Ｈ－Ｆ００ＡＨ

ＣＭＴ用ワークエリア.

Ｆ００ＢＨ　ターミナルモード用ワークエリア.

Ｆ００ＣＨ　ファンクションキー・ナンバ.

Ｆ００ＤＨ－Ｆ０１２Ｈ

タイマ用バッファ．ＢＣＤコードで秒，分，時，日，月，年の順.

Ｆ０１３Ｈ－Ｆ０ＣＦＨ

グラフィック関係用ワークエリア.

Ｆ０Ｄ０Ｈ－Ｆ０Ｄ２Ｈ

未使用.

Ｆ０Ｄ３Ｈ－Ｆ１５２Ｈ

ＵＳＡＲＴ用入力キュー.

Ｆ１５３Ｈ，５４Ｈ

ＲＳ－２３２Ｃ入出力用フラグ.

Ｆ１５５Ｈ－Ｆ１６ＢＨ

モニタ・メインＲＯＭコールルーチン.

```
CALL   0F155H
DEFW   <ADDRESS>
```

でＣＡＬＬ.

Ｆ１６ＣＨ－Ｆ１８３Ｈ

モニタＲＯＭセレクト用サブルーチン.

Ｆ１７５Ｈ－Ｆ１８３Ｈ

メインＲＯＭセレクト用サブルーチン．

Ｆ１８４Ｈ－Ｆ１８９Ｈ

モニタ・ブロック転送用サブルーチン．

Ｆ１８ＡＨ－Ｆ１Ｂ４Ｈ

モニタ・１バイト転送用サブルーチン．

メインＲＯＭ………Acc＝００Ｈ

モニタＲＯＭ………Acc＝０１Ｈ

Sub　ＲＯＭ………Acc＝０２Ｈ

Acc →（０Ｆ１Ｅ１Ｈ）

ＨＬレジスタ＝アドレス

としてＣＡＬＬする．結果は Acc に得られる．

Ｆ１Ｂ５Ｈ－Ｆ１ＢＢＨ

Accの内容をメインＲＡＭに転送するサブルーチン．

Ｆ１ＢＣＨ－Ｆ１Ｃ１Ｈ

モニタＲＯＭをセレクトしエラー処理ルーチンに制御を移す．

Ｆ１Ｃ２Ｈ－Ｆ１ＣＤＨ

モニタ・Ｇコマンド用サブルーチン．

Ｆ１ＣＥＨ－Ｆ２１ＤＨ

モニタ・ワークエリア．

Ｆ２１ＥＨ－Ｆ２ＦＦＨ

未使用．

Ｆ３００Ｈ－Ｆ３１ＦＨ

インタラプト・ベクタ．

Ｆ３２０Ｈ－Ｆ３Ｃ７Ｈ

未使用．

Ｆ３Ｃ８Ｈ－ＦＦＦ７Ｈ

テキスト用ビデオＲＡＭ．

ＦＦＦ８Ｈ－ＦＦＦＦＨ

未使用．

4－2　N88－DISK・BASIC　フックアドレス一覧.

N88－DISK　BASIC特有の動作を行わせるためのジャンプテーブルで，飛び先アドレスの指定の無いものは全てRET命令のみが書き込まれています.

ECBBH → 8A12H　ディスクの諸元のセット.
ECBEH → 　未使用.
ECC1H → A736H　REMOVE処理を行う.
ECC4H → 9BC5H　ディスクへのSAVEを行う.
ECC7H → 　未使用.
ECCAH → 　未使用.
ECCDH → 8E58H　IDセクタを実行する.
ECD0H → 9FFDH　ディスクファイルに対するGET#／PUT#を行う.
ECD3H → 91A5H　EDIT用行番号をインクリメントする.
ECD6H → 91BEH　EDIT用行番号をデクリメントする.
ECD9H → 9C51H　ディスクからのLOADを行う.
ECDCH → A6E3H　LINE　INPUT#処理用.
ECDFH → 　未使用.
ECE2H → 9118H　ROLL　DOWNキーの処理を行う.
ECE5H → 90CCH　ROLL　UPキーの処理を行う.
ECE8H → A08DH　ディスクへのシーケンシャル出力を行う.
ECEBH → 　未使用.
ECEEH → A0D4H　ディスクからのシーケンシャル入力を行う.
ECF1H → 　未使用.
ECF4H → 　未使用.
ECF7H → 　未使用.
ECFAH → 　未使用.
ECFDH → 　未使用.
ED00H → 　未使用.
ED03H → 　未使用.
ED06H → 8E55H　デバイス・ナンバのデフォルト値をセットする.
ED09H → 8E9AH　デバイス・ナンバを得る.
ED0CH → 　未使用.
ED0FH → 8F0AH　プログラム実行用.

ED12H →　　　　未使用.
ED15H →　　　　未使用.
ED18H →　　　　未使用.
ED1BH →　　　　未使用.
ED1EH →　　　　未使用.
ED21H →　　　　未使用.
ED24H → 8DE5H　CHAIN用行入力処理.
ED27H → 8FA4H　DSKI＄，INPUT＄，ATTR＄への分岐.
ED2AH →　　　　未使用.
ED2DH → 8FBDH　EOF，LOC，FPOS，LOFへの分岐.
ED30H →　　　　未使用.
ED33H →　　　　未使用.
ED36H →　　　　未使用.
ED39H →　　　　未使用.
ED3CH → 8F06H　LIST用.
ED3FH →　　　　未使用.
ED42H → 91E3H　出力用.
ED45H →　　　　未使用.
ED48H →　　　　未使用.
ED4BH →　　　　未使用.
ED4EH →　　　　未使用.
ED51H →　　　　未使用.
ED54H →　　　　未使用.
ED57H →　　　　未使用.
ED5AH → 8F7CH　倍精度関数への分岐.
ED5DH →　　　　未使用.
ED60H →　　　　未使用.
ED63H →　　　　未使用.
ED66H → 8FDDH　フルセンテンスエラーメッセージ出力用.
ED69H →　　　　未使用.
ED6CH →　　　　未使用.
ED6FH → 9B21H　ディスクファイルをCLOSEする.
ED72H →　　　　未使用.
ED75H → 8F41H　倍精度型式評価用.

ＥＤ７８Ｈ →　　　　　未使用.
ＥＤ７ＢＨ →　　　　　未使用.
ＥＤ７ＥＨ →　　　　　未使用.
ＥＤ８１Ｈ →　　　　　未使用.
ＥＤ８４Ｈ →　　　　　未使用.
ＥＤ８７Ｈ →　　　　　未使用.
ＥＤ８ＡＨ →　　　　　未使用.
ＥＤ８ＤＨ →　　　　　未使用.
ＥＤ９０Ｈ →　　　　　未使用.
ＥＤ９３Ｈ →　　　　　未使用.
ＥＤ９６Ｈ →　　　　　未使用.
ＥＤ９９Ｈ →　　　　　未使用.
ＥＤ９ＣＨ →　　　　　未使用.
ＥＤ９ＦＨ → ＡＤ４８Ｈ　プログラムプロテクト・チェックルーチン.
ＥＤＡ２Ｈ → ＡＣＣ１Ｈ　ＳＡＶＥ文Ｐオプションセーブ用.
ＥＤＡ５Ｈ → ＡＤ０５Ｈ　ＬＯＡＤ文Ｐオプションロード用.
ＥＤＡ８Ｈ → ＡＤ５１Ｈ　プログラムプロテクト・チェックルーチン.
ＥＤＡＢＨ → ９Ａ０１Ｈ　ディスクファイルをＯＰＥＮする.
ＥＤＡＥＨ →　　　　　未使用.
ＥＤＢ１Ｈ → ９９４ＢＨ　ドライブバッファ・アドレスセット.
ＥＤＢ４Ｈ →　　　　　未使用.
ＥＤＢ７Ｈ →　　　　　未使用.
ＥＤＢＡＨ →　　　　　未使用.
ＥＤＢＤＨ →　　　　　未使用.
ＥＤＣ０Ｈ →　　　　　未使用.
ＥＤＣ３Ｈ →　　　　　未使用.
ＥＤＣ６Ｈ →　　　　　未使用.
ＥＤＣ９Ｈ → ８Ｅ４ＢＨ　ＣＨＡＩＮ文エラー処理用.
ＥＤＣＣＨ →　　　　　未使用.
ＥＤＣＦＨ →　　　　　未使用.
ＥＤＤ２Ｈ →　　　　　未使用.
ＥＤＤ５Ｈ →　　　　　未使用.
ＥＤＤ８Ｈ →　　　　　未使用.
ＥＤＤＢＨ →　　　　　未使用.

ＥＤＤＥＨ → 未使用.
ＥＤＥ１Ｈ → 未使用.
ＥＤＥ４Ｈ → 未使用.
ＥＤＥ７Ｈ → 未使用.
ＥＤＥＡＨ → 未使用.
ＥＤＥＤＨ → 未使用.
ＥＤＦ０Ｈ → 未使用.
ＥＤＦ３Ｈ → 未使用.
ＥＤＦ６Ｈ → ９０８ＢＨ ＤＭＡタイプ・サブルーチン.
ＥＤＦ９Ｈ → 未使用.
ＥＤＦＣＨ → 未使用.
ＥＤＦＦＨ → ９９７７Ｈ ドライブ・マウントのチェック.
ＥＥ０２Ｈ → 未使用.
ＥＥ０５Ｈ → 未使用.
ＥＥ０８Ｈ → 未使用.
ＥＥ０ＢＨ → 未使用.

4－3 N₈₈－DISK BASIC 用ファイル処理一覧.

N₈₈－DISK BASICでサポートされたファイル処理を行わせるためのジャンプテーブルです．次のアドレスが書き込まれていた場合には各々のエラーが発生します．

アドレス４ＤＣ１Ｈ：Feature not available
アドレス０Ｂ０６Ｈ：Illegal function call

〔COM2：，COM3：〕

ＥＥ０ＥＨ → ４ＤＣ１Ｈ　ＯＰＥＮ
ＥＥ１１Ｈ → ４ＤＣ１Ｈ　ＣＬＯＳＥ
ＥＥ１４Ｈ → ４ＤＣ１Ｈ　ＰＵＴ#，ＧＥＴ#
ＥＥ１７Ｈ → ４ＤＣ１Ｈ　ＯＵＴＰＵＴ
ＥＥ１ＡＨ → ４ＤＣ１Ｈ　ＩＮＰＵＴ
ＥＥ１ＤＨ → ４ＤＣ１Ｈ　ＬＯＣ
ＥＥ２０Ｈ → ４ＤＣ１Ｈ　ＬＯＦ
ＥＥ２３Ｈ → ４ＤＣ１Ｈ　ＥＯＦ
ＥＥ２６Ｈ → ４ＤＣ１Ｈ　ＦＰＯＳ
ＥＥ２９Ｈ → ４ＤＣ１Ｈ　Data read
ＥＥ２ＣＨ → ４ＤＣ１Ｈ　ＷＩＤＴＨ　# n

〔SCRN：〕

ＥＥ２ＦＨ → ９０４３Ｈ　ＯＰＥＮ
ＥＥ３２Ｈ → ４８３ＤＨ　ＣＬＯＳＥ
ＥＥ３５Ｈ → ９０５ＦＨ　ＰＵＴ#，ＧＥＴ#
ＥＥ３８Ｈ → ９０５１Ｈ　ＯＵＴＰＵＴ
ＥＥ３ＢＨ → ０Ｂ０６Ｈ　ＩＮＰＵＴ
ＥＥ３ＥＨ → ０Ｂ０６Ｈ　ＬＯＣ
ＥＥ４１Ｈ → ０Ｂ０６Ｈ　ＬＯＦ
ＥＥ４４Ｈ → ０Ｂ０６Ｈ　ＥＯＦ
ＥＥ４７Ｈ → ０Ｂ０６Ｈ　ＦＰＯＳ
ＥＥ４ＡＨ → ０Ｂ０６Ｈ　Read data
ＥＥ４ＤＨ → ０Ｂ０６Ｈ　ＷＩＤＴＨ　# n

〔KYBD:〕

EE50H → 9001H　OPEN
EE53H → 483DH　CLOSE
EE56H → 9018H　PUT#, GET#
EE59H → 0B06H　OUTPUT
EE5CH → 900CH　INPUT
EE5FH → 903CH　LOC
EE62H → 0B06H　LOF
EE65H → 0B06H　EOF
EE68H → 0B06H　FPOS
EE6BH → 9032H　Read data
EE6EH → 0B06H　WIDTH #n

第5章　PC-8031とのインタフェース構造

——Interface Structure With PC-8031——

5－1　ＰＣ－８０３１－２Ｗとのインタフェース構造．

5－1－1　ＰＣ－８０３１－２Ｗについて

ＰＣ－８０３１－２Ｗは本体（ここではＰＣ－８８０１）とは独立したＣＰＵ，ファームウェアを持ったインテリジェントタイプのミニフロッピィディスク・ユニットです．ＣＰＵにはＺ－８０ＡとコンパチブルなμＰＤ７８０Ｃが搭載され，ディスクドライブの制御には高性能なフロッピディスク・コントローラ（ＦＤＣ）であるμＰＤ７６５，また本体側とのインタフェースには汎用パラレルインタフェース用ＬＳＩ・μＰＤ８２５５が使用されています．

また，ＰＣ－８０３１－２Ｗには両面仕様のドライブユニットが搭載されていますが，ＰＣ－８０３１とのコンパチビリティを考慮して片面仕様としても使用できるようなコマンドも用意されています．それに，ＰＣ－８０３１－２Ｗで強化されたものに２バイト転送機能があり，４つのコマンドでそれを使用可能です．あと，ＰＣ－８０３１－２Ｗ内でユーザプログラムを走らせる際に有用なさまざまなコマンドが用意されています．これらのコマンドを有効に利用すればこのタイプのディスクユニットの欠点である速度の点をカバーすることさえも可能です．（最近のＤＯＳ（ディスク・オペレーティング・システム）はディスクユニット内に独自のソフトウェアを作成し，それによる高速処理を実現しているものも登場してきています．）

ファームウェアはメモリ空間の００００Ｈ～０７ＦＦＨにＲＯＭとして焼きこまれており，この中にＦＤＣの制御ルーチンやコマンド処理ルーチンが入っています．またバッファとして８Ｋバイト，フリーエリアとして7.85Ｋバイト，ワークエリアとして0.05ＫバイトのＤＲＡＭが用意されていますので，ユーザはこの上で独自のプログラムを走らせることができます．次にＰＣ－８０３１－２Ｗ内のメモリマップをあげておきますので，ユーザプログラム作成の際の参考にして下さい．

〔ファームウェア0H～7FFH〕

0000H　システムリセット.

004EH　ブレークポイント処理.

007EH　システムスタート・ワークエリアイニシァライズ.

00C1H　コマンド入力待ち.

011BH　コマンド処理サブルーチン・アドレステーブル，２バイト×３１組.

0159H　コマンド＃10処理．ポート出力.

015DH　コマンド＃０処理．イニシァライズ.

01E7H　フォーマット用サブルーチン.

0263H　ＦＤＣよりレザルトステータスを受信し，ワークエリアにセットする．ＳＴ０・ＳＴ１・ＳＴ２・Ｃ・Ｈ・Ｒ・Ｎの順.

029AH　ＦＤＣからのデータを受信する.

02A4H　ＦＤＣへのデータを送信する.

02B4H　ディレイ・ルーチン.

02C2H　ＦＤフォーマット・メインルーチン.

02D5H　ＦＤリード／ライトルーチン.

035BH　マージンコントロール用.

0375H　マージンコントロール用.

0396H　ＷＲＩＴＥ　ＤＥＬＥＴＥＤ　ＤＡＴＡ（ＦＤＣ）コマンド実行.

03E9H　ＳＥＮＳＥ　ＤＥＶＩＣＥ　ＳＴＡＴＵＳ（ＦＤＣ）コマンド実行.

03F1H　ＦＤＣへのパラメータ送信.(ＨＤ・ＵＳ１・ＵＳ０，Ｃ，Ｈ，Ｒ，Ｎ，ＥＯＴ，ＧＰＬ，ＤＴＬの順)

040EH　ＨＤ・ＵＳ１・ＵＳ０合成バイトを得る．（Creg：デバイス，D reg：トラック)

041BH　シリンダ番号を得る．（D reg：トラック)

0423H　指定デバイスのモード（ＤＳ／ＳＳ）を得る.（C reg：デバイス)

042FH　ヘッド番号を得る.（D reg：トラック)

0438H　コマンド＃４処理．コピー.

045EH　コマンド＃１処理．ＦＤへの書き込み.

045FH　コマンド＃17処理．高速受信を行う.

048CH　コマンド＃２処理．ＦＤからの読み出し.

04DDH　コマンド＃３処理．リードバッファのデータを送信.

04DEH　コマンド＃18処理．高速送信を行う.

050BH　コマンド＃６処理．エラーステータスを送信.

0511H　コマンド＃７処理．ドライブの接続状況を送信．
0517H　コマンド＃５処理．ドライブをフォーマットする．
052FH　コマンド＃11処理．メモリ上のデータを送信
053BH　コマンド＃12処理．受信データをメモリにロード．
0547H　コマンド＃21処理．メモリ上のデータを高速送信．
0553H　コマンド＃22処理．高速受信データをメモリにロード．
056CH　コマンド＃13処理．制御を指定アドレスに移す．
0571H　コマンド＃14処理．ディスクよりデータをロードする．
057FH　コマンド＃15処理．データをディスクにセーブする．
058EH　コマンド＃16処理．ディスク内のプログラムを実行．
05A9H　コマンド＃19処理．レザルトステータス送信．
05B4H　コマンド＃20処理．レザルトステータス（ＳＴ３）送信．
05BCH　コマンド＃23処理．デバイスのモード設定．
05CEH　コマンド＃24処理．各デバイスのモードを送信．
05D4H　コマンド＃25処理．リードアフタライトモードにセット．
05D5H　コマンド＃26処理．リードアフタライトモードのリセット．
05DAH　コマンド＃27処理．ユーザプログラムの再開．
0602H　コマンド＃28処理．ブレークポイントセット
0617H　ブレークポイント設定チェック，ブレークポイント復帰．
0628H　コマンド＃29処理．レジスト値のセット．
0629H　コマンド＃30処理．レジスタ値の送信．
0649H　コマンド＃８処理．メモリのテストを行う．
0693H　コマンド＃９処理．メモリ上のデータを送信．
06A6H　４バイトデータ受信．（セクタ数，ドライブ，トラック，セクタ）
06AFH　２バイトデータ受信．（アドレス）
06B4H　パラメータの照合．
06C9H　書き込み補償制御．
06D9H　１バイトデータ受信．
070FH　１バイトデータ送信．
0741H　１バイトデータ受信・メモリ格納．
077BH　メモリ上の１バイトデータを送信．
07B1H　メモリ上の２バイトデータを送信．
07BCH　２バイトデータ受信・メモリ格納．
07C6H　割り込み受付後，一定の時間待ちを行う（ＰＣ－８０Ｓ３１のみ）．

07D6H　イニシャライズ後，FDCのステータス(ST3）のTSビット(第3ビット）が1になるまでループを行う．

07F0H　マージンデータ領域（4バイト×4組)

〔ライトバッファ4000H～4FFFH〕
〔リードバッファ5000H～5FFFH〕
〔ユーザエリア　6000H～7EFFH〕
〔ワークエリア　7F00H～7FFFH〕

7F00H　マージンデータ格納アドレス．4デバイス分．

7F08H　セクタ数待避用．

7F09H　F8Hに出力したデータ．

7F0AH　エラーカウンター．

7F0BH　F7Hに出力したデータ．

7F0CH　レザルトステータス．8バイト．

7F14H　エラーステータス．

7F15H　ドライブステータス．

7F16H　コマンド受信中フラグ．

7F17H　パラメータセット済フラグ．

7F18H　ドライブ番号待避用．

7F19H　トラック番号待避用．

7F1AH　セクタ番号待避用．

7F1BH　モード設定状況．

7F1CH　各ドライブのモード．4デバイス分．

7F20H　リードアフタライトモード設定フラグ．

7F21H　FDCのリード／ライトモード．

7F22H　ブレークポイント処理用ジャンプテーブル(I)．未使用．

7F25H　ブレークポイント処理用ジャンプテーブル(II)．

7F28H　レジスタ待避用．（SP，PC，I，IY，IX，HL′，DE′，BC′，AF′，HL，DE，BC，AFの順)

7F42H　ブレークポイント設定フラグ．

7F43H　ブレークポイントアドレス．

7F45H　ブレークポイントデータ待避用．

5—1—2　ハンドシェイクについて

ＰＣ－８０３１－２Ｗは前述の様に本体側（ＰＣ－８８０１）とのインタフェースに汎用パラレルインタフェースＬＳＩ，μＰＤ８２５５を使用しており，ＰＣ－８８０１にも同じくμＰＤ８２５５がミニフロッピィディスクとのインタフェース用に使用されています．そして，これらの間のデータのやりとりには３線ハンドシェイクという方法が用いられており，このとき８２５５の各ポートは次の様に設定及びＩ／Ｏアドレスの割り当てが行われています．

ポートＡ　全ビット　　——データ入力用　　・・・ＦＣＨ
ポートＢ　全ビット　　——データ出力用　　・・・ＦＤＨ
ポートＣ　上位４ビット——制御データ出力用・・・ＦＥＨ
ポートＣ　下位４ビット——制御データ入力用・・・ＦＦＨ

これらのポートの設定は本体側，８０３１側で共通で，結線のようすは下図（Fig. ５－１－１）のとおりです．

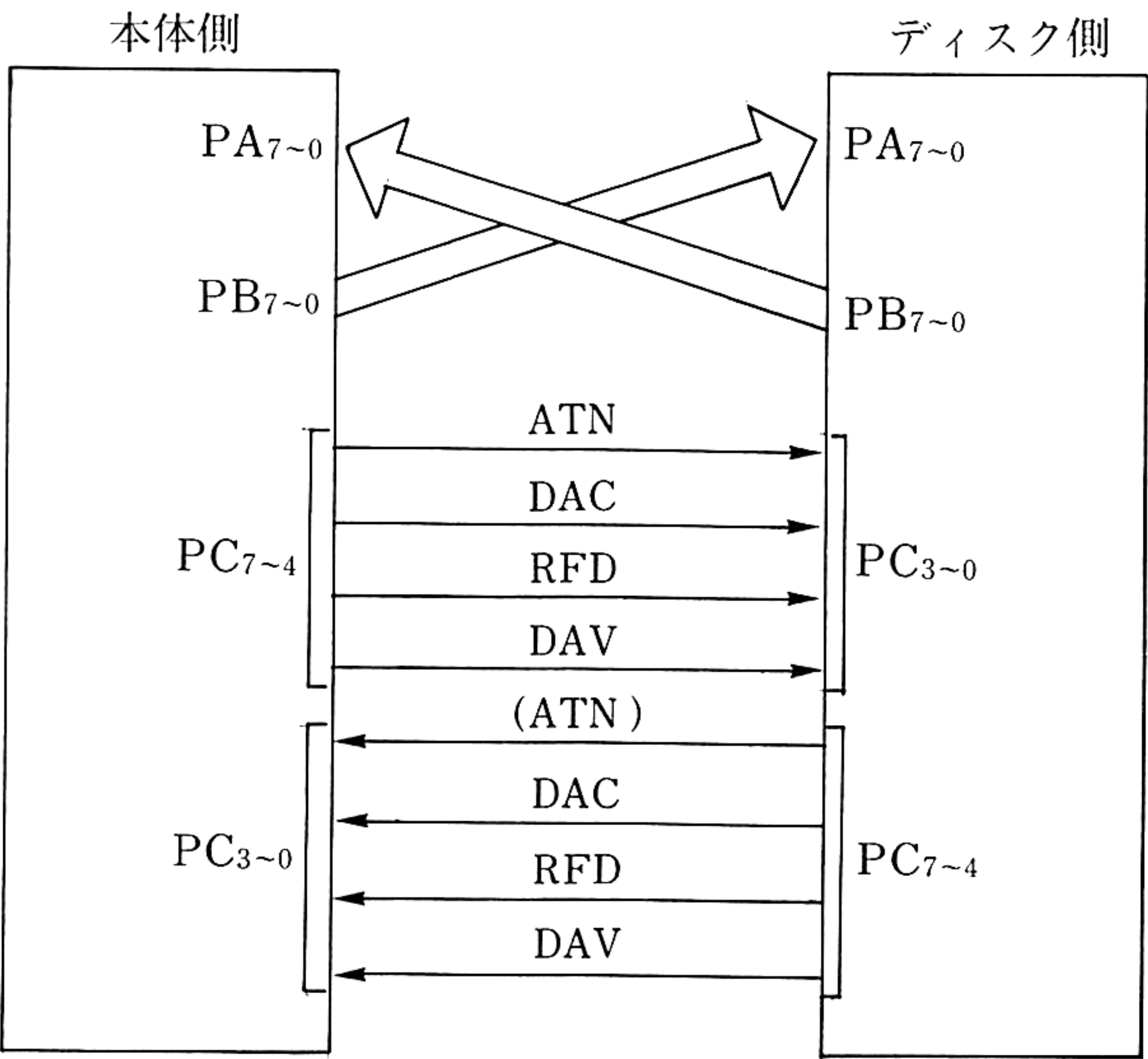



Fig.５－１－１　各ポートの結線状況

以上のことからわかるように，８２５５をモード０で使用している訳です．このモードへの設定はＩ／ＯアドレスＦＦＨにコントロールワード９１Ｈを出力することで実現できます（Ｉ／ＯアドレスＦＦＨはポートＣへの出力用にも割り当てられていることに注意して下さい）．

5－1－3　制御データについて

前項でポートAはデータの入力用に，ポートBはデータ出力用に設定されていると述べましたが，これらの入出力はポートCの各制御ビットを制御することで行わすことができます．以下に各制御ビットの名称と役割をまとめて示します．

第7ビット（ATN：Attention）・・・本体側からのデータ出力の際に，そのデータがコマンドであることを示します．

第6ビット（DAC：Data accepted）・・・本体側でデータの受け取りが終了していることを示します．

第5ビット（RFD：Ready for data）・・本体側でデータの受け取りの準備ができていることを示します．

第4ビット（DAV：Data valid）・・・本体側から出力したデータが有効であることを示します．

第3ビット・・・このビットはPC－8031－2Wがディスクユニットとして動作している限り意味を持ちません．

第2ビット（DAC：Data accepted）・・・ディスクユニット側でデータを受け取ったことを示します．

第1ビット（RFD：Ready for data）・・ディスクユニット側でデータ受け取りの準備ができていることを示します．

第0ビット（DAV：Data valid）・・・ディスクユニット側で送ったデータが有効であることを示します．

ここで，第7～4ビットは制御データ出力用，第3～0ビットは制御データ入力用です．また，第3～0ビットはI／OアドレスFEHから直接読み取ることができますが，第7～4ビットはI／OアドレスFFHに直接出力が可能ということにはなりません．これは前述の様にI／OアドレスFFHが8255のモード設定用にも割り当てられているためで，実際のポートCへの出力は次の様に行うことになっています．以下にI／OアドレスFFHに出力するデータの各ビットの意味を示します．

第7ビット・・・このビットがセットされている場合に8255はモード設定を選択します．8255をモード0に設定したい場合にはI／OアドレスFFHに91Hを出力すればよい訳です．

第3～1ビット・・・これら3ビットによってポートCの8ビットのうちどのビットに対してセット，リセットを行うか指定します．

第0ビット・・・第3～1ビットで指定したビットを，セットするかリ

セットするかを指定します．

では次に，これらのビットの動きを見ながら1バイトのデータを送受信するやりとりを見てみます（実際は一度に2バイト送受信することが可能ですが，ここでは1バイトの場合を例にとります）．

〔データ送信の場合〕

1) ディスク側から制御データを入力し，ＲＦＤビットがセットされるまで待ちます．
2) 出力データをポートＢに出力します．
3) ＤＡＶビットをセットし，出力したデータが有効であることを示します．
4) ディスク側から制御データを入力し，ＤＡＣビットがセットされるまで待ちます．
5) ＤＡＶビットをリセットします．
6) ディスク側から制御データを入力し，ＤＡＣビットがリセットされるまで待ちます．
7) データ送信終了．

〔データ受信の場合〕

1) ＲＦＤビットをセットし，データ受け取りの準備ができていることを示します．
2) ディスク側から制御データを入力し，ＤＡＶビットがセットされるまで待ちます．
3) ＲＦＤビットをリセットします．
4) ポートＡよりデータを入力します．
5) ＤＡＣビットをセットし，データを受け取ったことを示します．
6) ディスク側から制御データを入力し，ＤＡＶビットがリセットされるまで待ちます．
7) ＤＡＣビットをリセットします．
8) データ受信終了．

コマンド送信の場合は，データ送信におけるＲＦＤビットのセット待ちの前後にそれぞれＡＴＮビットのセット，ＡＴＮビットのリセットを挿入するだけで，あとは共通です．

また，一度のハンドシェイクで2バイトのデータを送受信できると述べましたが，これは次のようにすれば可能です．

〔データ送信の場合〕

5)と6)の間にもう一度データをポートBに出力する.

〔データ受信の場合〕

6)と7)の間にもう一度データをポートAより入力する.

これはDAV，DACビットの役割を逆に利用したもので，2回目のデータの入出力はこれらのビットがリセットされることで意味を持つ訳です.つまり，DAVビットのリセットにより送信側は2回目のデータが有効であることを受信側に示します.受信側はさらにデータ入力後DACビットをリセットして2回目のデータを受け取ったことを送信側に知らせる，という具合です.

N88－BACICでは1バイトデータ，2バイトデータの送受信の両方の機能を備えたサブルーチンが用意されており，これらはコマンドの種類によって使い分けなければなりません.詳しくは第3章及び5－2を御参照下さい.

最後に，2バイトデータ送受信の場合のフローチャートをFig.5－2－2として掲げておきます.

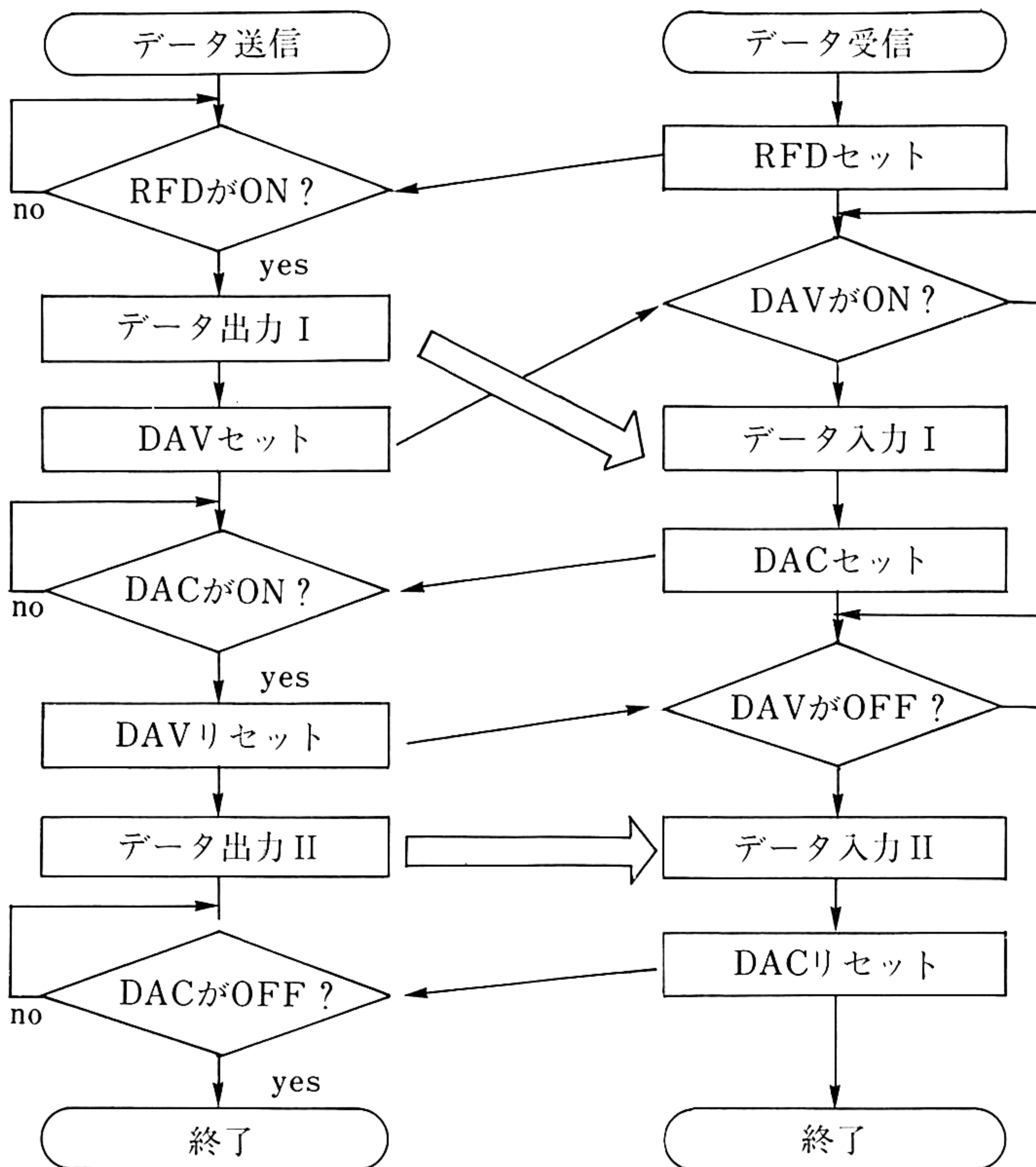



Fig 5 − 2 − 2　2バイトデータ送受信

5－2　ＰＣ－８０３１－２Ｗ（ＰＣ－８０Ｓ３１）のコマンド一覧

本節では，ＰＣ－８０３１－２Ｗ（ＰＣ－８０Ｓ３１）の持つ３１種のコマンドを紹介していきます．これらのうち，コマンドナンバ１７以降はＰＣ－８０３１にない新しく追加されたコマンドです．また，コマンドナンバ１７，１８，２１，２２は１度に２バイトずつデータを送受する仕様のコマンドですので，N88－ＢＡＳＩＣ－ＲＯＭ内のサブルーチンを利用する場合には，ＥＦ１５Ｈ番地に００Ｈ以外の値が書き込まれている必要があります．詳しくは第３章の各サブルーチンの説明を御参照下さい．

〔０（００Ｈ）〕

イニシァライズ・コマンド

送信フォーマット：コマンドコード００Ｈ

受信フォーマット：――――――――――

ＰＣ－８０３１（－２Ｗ）のモータをスタートさせ，ディスクユニットのイニシァライズを行うコマンドです．

〔１（０１Ｈ）〕

ライトディスク・コマンド

送信フォーマット：コマンドコード０１Ｈ
書き込むセクタ数
ドライブナンバ（０～）
トラック・ナンバ
セクタ・ナンバ
データ群（書き込むセクタ数×２５６バイト）

受信フォーマット：――――――――――

本体より，まとまったデータを直接ディスクに書き込むためのコマンドです．ただし，２つのトラックにまたがって書き込むことはできません．

〔2（02H）〕

リードディスク・コマンド

送信フォーマット：コマンドコード０２Ｈ
　　　　　　　　　読み出すセクタ数
　　　　　　　　　ドライブ・ナンバ（０～）
　　　　　　　　　トラック・ナンバ
　　　　　　　　　セクタ・ナンバ
受信フォーマット：――――――――――

ディスク上のデータをＰＣ－８０３１（－２W）内のリードバッファに読み出すためのコマンドです．通常はコマンド・ナンバ3のセンドデータ・コマンドとペアで使用されます．ただし，２つのトラックにまたがって読み出すことはできません．

〔3（03H）〕

センドデータ・コマンド

送信フォーマット：コマンドコード０３Ｈ
受信フォーマット：データ群（読み込むセクタ数×２５６バイト）

ＰＣ－８０３１（－２W）のリードバッファからデータを本体へ転送させるコマンドです．通常このコマンドはコマンドナンバ2とペアで使用され，データの数はコマンドナンバ2でリードバッファに読み出したデータの数と同一です．

〔4（04H）〕

コピー・コマンド

送信フォーマット：コマンドコード０４Ｈ
　　　　　　　　　転送するセクタ数
　　　　　　　　　転送元のドライブ・ナンバ
　　　　　　　　　転送元のトラック・ナンバ
　　　　　　　　　転送元のセクタ・ナンバ

転送先のドライブ・ナンバ
転送先のトラック・ナンバ
転送先のセクタ・ナンバ
受信フォーマット：――――――――――

セクタ間のコピーを行うコマンドです．ここで，２つのトラックにまたがってセクタ数を指定することはできません．

〔５（０５Ｈ）〕
フォーマット・コマンド

送信フォーマット：コマンドコード０５Ｈ
ドライブ・ナンバ（０～）
受信フォーマット：――――――――――――

指定されたドライブ内のディスクをフォーマッティングするコマンドです．

〔６（０６Ｈ）〕
レザルト・ステータス・コマンド

送信フォーマット：コマンドコード０６Ｈ
受信フォーマット：実行結果

各コマンドの実行結果を得るコマンドです．そのとき，実行結果の各ビットは次のような意味をもちます．

7　6　5　4　3　2　1　0

エラーの発生した時にセット
リードバッファにデータのある時にセット
I／O動作終了済フラグ

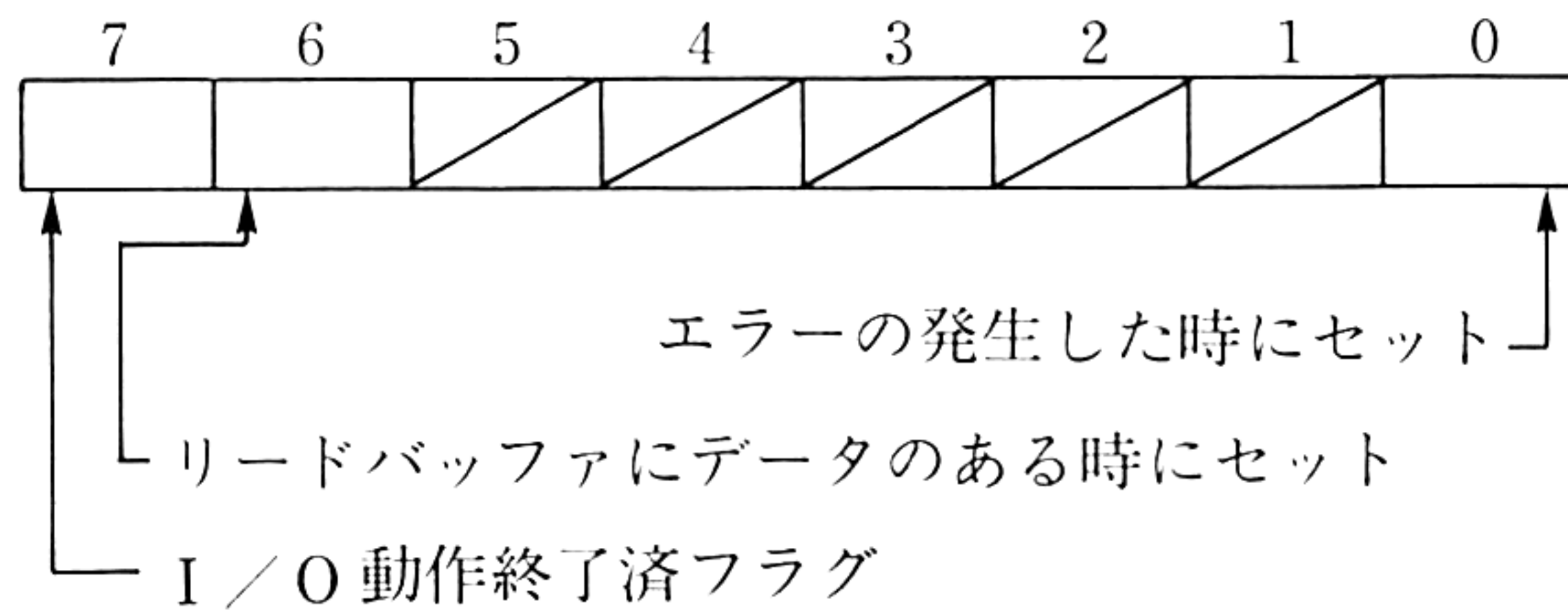



Fig５－２－１　レザルトステータス・バイト

第６ビットが０である（リードバッファにデータがない）ときにセンドデータ・コマンドを実行すると無限ループに入りこみますので注意が必要です．

〔7（07H）〕
ドライブ・ステータス・コマンド

送信フォーマット：コマンドコード07H
受信フォーマット：ドライブ・ステータス

ドライブの接続状況を本体へ転送するためのコマンドです．そのとき，各ビットの意味は次の様になります．

7	6	5	4	3	2	1	0
				1	1	1	1
ドライブ接続フラグ				ディスクセットフラグ			
3	2	1	0	3	2	1	0

FDCに対するデバイス・ナンバ

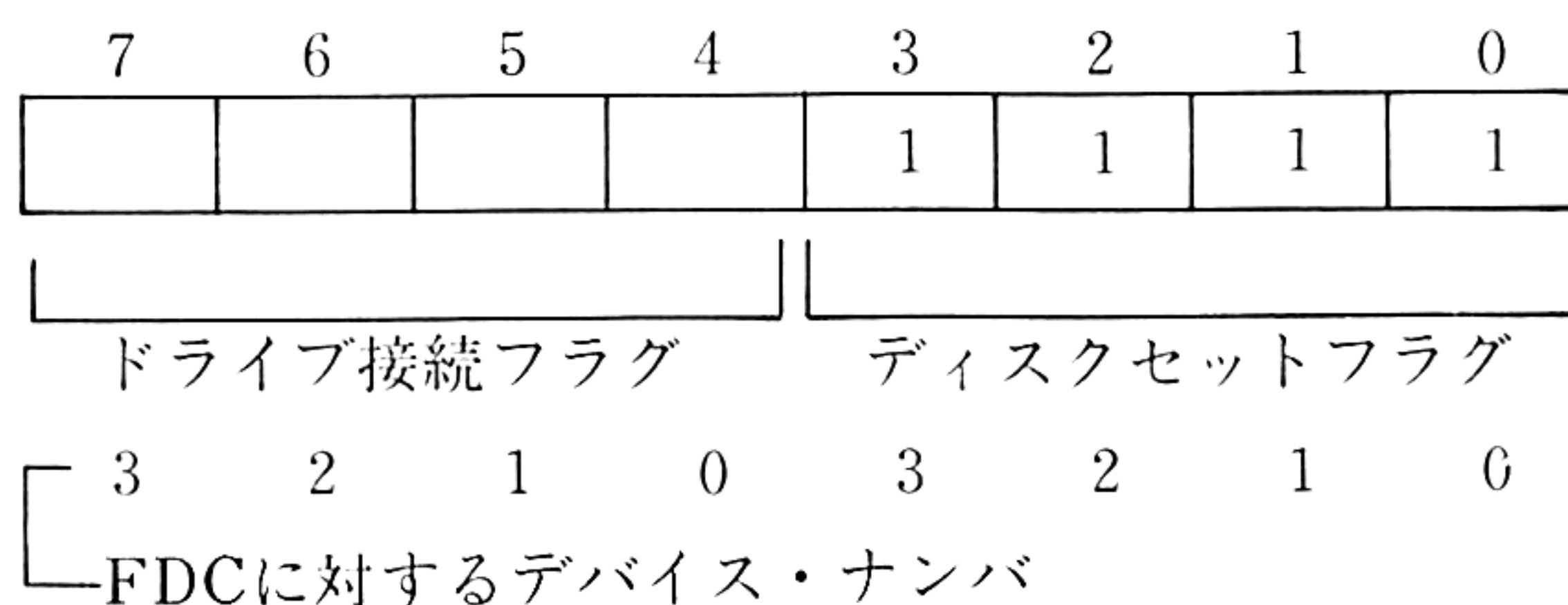



Fig5－2－2　ドライブステータス・バイト

ここで，ディスク・セット・フラグは常に1にセットされており意味を持ちません．

〔8（08H）〕
テストメモリ・コマンド

送信フォーマット：コマンドコード08H
受信フォーマット：ヘッダ（81H）
エラーアドレス（上位）
エラーアドレス（下位）
書き込まれたデータ
読み出された異常データ
デリミタ（80H）

ここで，受信データはエラーの発生した回数だけ同じパターンがくり返されます．

〔9（09H）〕
センド・メモリデータ・コマンド

送信フォーマット：コマンドコード０９Ｈ
　　　　　　　　　転送アドレス（上位）
　　　　　　　　　転送アドレス（下位）
　　　　　　　　　転送バイト数（上位）
　　　　　　　　　転送バイト数（下位）
受信フォーマット：データ数（転送バイト数）

ＰＣ－８０３１－２Ｗのメモリ上のデータを本体に転送するためのコマンドです．

〔１０（０ＡＨ）〕
ポート・アウトプット・コマンド

送信フォーマット：コマンドコード０ＡＨ
　　　　　　　　　出力データ
受信フォーマット：――――――――――

ＰＣ－８０３１－２Ｗ内の出力ポートＦ７Ｈ番地に１バイトのデータを出力するためのコマンドです．出力ポートＦ７Ｈ番地はプリンタの制御用（但しＰＣ－８０３１，ＰＣ－８０３１－２Ｗのみ）と，フロッピィディスク・コントローラ（ＦＤＣ）の制御用に用いることができます．

〔１１（０ＢＨ）〕
センド・メモリデータ・コマンド

送信フォーマット：コマンドコード０ＢＨ
　　　　　　　　　転送アドレス（上位）
　　　　　　　　　転送アドレス（下位）
　　　　　　　　　転送バイト数（上位）
　　　　　　　　　転送バイト数（下位）
受信フォーマット：データ群（転送バイト数）

コマンドナンバ９と同様です．

〔１２（０ＣＨ）〕

レシーブ・メモリデータ・コマンド

送信フォーマット：コマンドコード０ＣＨ
転送アドレス（上位）
転送アドレス（下位）
転送バイト数（上位）
転送バイト数（下位）
データ群（転送バイト数）
受信フォーマット：――――――――――

本体よりＰＣ－８０３１－２Ｗのメモリ上にデータを転送するためのコマンドです．

〔１３（０ＤＨ）〕

エグゼキュート・コマンド

送信フォーマット：コマンドコード０ＤＨ
実行アドレス（上位）
実行アドレス（下位）
受信フォーマット：――――――――――

ＰＣ－８０３１－２Ｗのメモリ上の指定されたアドレスに実行を移すためのコマンドです．不用意にこのコマンドを実行しますと暴走する場合がありますので注意が必要です．

〔１４（０ＥＨ）〕

ロードデータ・コマンド

送信フォーマット：コマンドコード０ＥＨ
セクタ数
ドライブ・ナンバ（０～）
トラック・ナンバ
セクタ・ナンバ

ロードアドレス（上位）
ロードアドレス（下位）
受信フォーマットット：――――――――――――

ディスク上のデータをＰＣ－８０３１－２Ｗの指定されたアドレスに読み出すためのコマンドです．

〔１５（０ＦＨ）〕
セーブデータ・コマンド

送信フォーマット：コマンドコード０ＦＨ
セクタ数
ドライブ・ナンバ（０～）
トラック・ナンバ
セクタ・ナンバ
セーブアドレス（上位）
セーブアドレス（下位）
受信フォーマット：――――――――――――

ＰＣ－８０３１－２Ｗの指定されたアドレス上のデータをディスクへ書き込むためのコマンドです．

〔１６（１０Ｈ）〕
ロードアンドゴー・コマンド

送信フォーマット：コマンドコード１０Ｈ
受信フォーマット：――――――――――――

ドライブ・ナンバ０，トラック・ナンバ０，セクタ・ナンバ１のデータをＰＣ－８０３１－２Ｗのメモリ５０００Ｈ番地からの領域にロードし，この５０００Ｈ番地に実行を移します．これを利用すると，ＩＰＬのような動作をコマンドで行わせることができます．

〔17（11H）〕

ライトディスク・コマンド（高速転送）

送信フォーマット：コマンドコード11H
セクタ数
ドライブ・ナンバ（0～）
トラック・ナンバ
セクタ・ナンバ
データ群（セクタ数×256バイト）
受信フォーマット：――――――――――

このコマンドは基本的な動作はコマンドナンバ1と同じですが，1度のハンドシェイクで2バイトのデータを送ることに注意して下さい．

〔18（12H）〕

センドデータ・コマンド（高速転送）

送信フォーマット：コマンドコード12H
受信フォーマット：データ群

このコマンドは基本的な動作はコマンドナンバ3と同じですが，1度のハンドシェイクで2バイト受け取ることに注意して下さい．

〔19（13H）〕

エラーステータス・コマンド

送信フォーマット：コマンドコード13H
受信フォーマット：エラーデータ0（FDCへ送ったコマンド）
エラーデータ1（ST0）
エラーデータ2（ST1）
エラーデータ3（ST2）
エラーデータ4（C：シリンダ・ナンバ）
エラーデータ5（H：ヘッド・ナンバ）
エラーデータ6（R：セクタ・ナンバ）
エラーデータ7（N：1セクタの長さ）

ＰＣ－８０３１内のフロッピィディスク・コントローラ（ＦＤＣ）のレザルト・ステータスの内容を本体へ転送するためのコマンドです．

各ステータスの詳しい内容についてはＦＤＣ（μＰＤ７６５）に関する資料を御参照下さい．

7	FT	………Fault信号の状態
6	WP	………Write Protect信号の状態
5	RY	………Ready信号の状態
4	T 0	………Track 0 信号の状態
3	TS	………Two Side信号の状態
2	HD	………Side Select信号の状態
1	US 1	………Unit Select 1 信号の状態
0	US 0	………Unit Select 0 信号の状態

Fig 5 ― 2 ― 3　デバイスステータス・バイト

〔２０（１４Ｈ）〕

デバイスステータス・コマンド

送信フォーマット：コマンドコード１４Ｈ
　　　　　　　　　ドライブ・ナンバ（０～）
受信フォーマット：デバイス・ステータス

ＰＣ－８０３１内のフロッピィディスク・コントローラ（ＦＤＣ）のレザルトステータスのステータス３（ＳＴ３）の内容を本体へ転送するためのコマンドです．ディスケットにプロテクトシールが貼られているかどうかをチェックする時に利用できます（ビット６をチェック）．

〔２１（１５Ｈ）〕

センド・メモリデータ・コマンド（高速転送）

送信フォーマット：コマンドコード１５Ｈ
　　　　　　　　　転送アドレス（上位）
　　　　　　　　　転送アドレス（下位）
　　　　　　　　　転送バイト数（上位）
　　　　　　　　　転送バイト数（下位）

受信フォーマット：データ群（転送バイト数）

このコマンドは基本的な動作はコマンドナンバ１１と同じですが，１度のハンドシェイクで２バイト受け取ることに注意して下さい．

〔２２（１６Ｈ）〕

レシーブ・メモリデータ・コマンド（高速転送）

送信フォーマット：コマンドコード１６Ｈ
転送アドレス（上位）
転送アドレス（下位）
転送バイト数（上位）
転送バイト数（下位）
データ群（転送バイト数）
受信フォーマット：―――――――――

このコマンドは基本的な動作はコマンドナンバ１２と同じですが，１度のハンドシェイクで２バイト送ることに注意して下さい．

〔２３（１７Ｈ）〕

モードチェンジ・コマンド

送信フォーマット：コマンドコード１７Ｈ
切り換え用データ
受信フォーマット：―――――――――

ＰＣ－８０３１－２Ｗの各ドライブを片面・両面モードのどちらかに設定するためのコマンドです．切り換え用データの各ビットの意味は次の様になります．

7	6	5	4	3	2	1	0
／	／	／	／				

FDCに対する　　　　　両面／$\overline{\text{片面}}$
デバイスナンバ　　　→3　2　1　0

Fig5－2－4　モードセット・バイト

ここで，上位4ビットは意味を持ちません．

〔24（18H）〕
センド・モードデータ・コマンド

送信フォーマット：コマンドコード18H
受信フォーマット：モードデータ

ＰＣ－８０３１－２Wの各ドライブが片面・両面のどちらのモードにセットされているかを知るためのコマンドです．各ビットの意味はコマンドナンバ23に準じます．

〔25（19H）〕
リードアフタライトモード・セット・コマンド

送信フォーマット：コマンドコード19H
受信フォーマット：――――――――

リードアフタライト・モードをセットするためのコマンドです．

〔26（1AH）〕
リードアフタライト・モードリセット・コマンド

送信フォーマット：コマンドコード1AH
受信フォーマット：――――――――

リードアフタライト・モードをリセットするためのコマンドです．

〔27（1BH）〕

リスタート・コマンド

送信フォーマット：コマンドコード１ＢＨ
受信フォーマット：――――――――――

コマンドナンバ２８で設定したブレークポイントにより中断したプログラムを再開させるためのコマンドです．

〔28（1CH）〕

ブレークポイント・セット・コマンド

送信フォーマット：コマンドコード１ＣＨ
　　　　　　　　　アドレス（上位）
　　　　　　　　　アドレス（下位）
受信フォーマット：――――――――――

ユーザプログラムにブレークポイントを設定するためのコマンドです．このコマンドを実行した場合，以前に設定したブレークポイントは無効となります．

〔29（1DH）〕

レジスタセット・コマンド

送信フォーマット：コマンドコード１ＤＨ
　　　　　　　　　レジスタ・ナンバ
　　　　　　　　　セットデータ（上位）
　　　　　　　　　セットデータ（下位）

ＰＣ－８０３１－２ＷのＣＰＵ内部レジスタにデータをセットするためのコマンドです．レジスタ・ナンバはそれぞれ次のレジスタを指します．

００Ｈ	ＡＦ	０７Ｈ	ＨＬ
０１Ｈ	ＢＣ	０８Ｈ	ＩＸ

02H	DE	09H	1Y
03H	HL	0AH	I
04H	AF′	0BH	PC
05H	BC′	0CH	SP
06H	DE′		

なお，I レジスタを指定した場合データは下位バイトのみ有効です．

〔30（1EH）〕
リードレジスタ・コマンド

送信フォーマット：コマンドコード1EH
　　　　　　　　　レジスタ・ナンバ
受信フォーマット：データ（上位）
　　　　　　　　　データ（下位）

ＰＣ－８０３１－２ＷのＣＰＵ内部レジスタの値を読み出すためのコマンドです．レジスタの指定はコマンドナンバ２９と同一です．

5－3　PC－80S31との相違

PC－80S31にはドライブの変更によると思われるソフトウェア上の変更点がいくつか見られます．そのほとんどがタイミングに関するものですが，以下にその主な変更点を揚げておきます．

1．FDCへのSPECIFYコマンドのパラメータで，SRT (Step Rate Time) の値が3から13に変更されている．
2．FDCのコントロールで，FDCからの割り込みを受けた後一定時間のループを行っている．
3．イニシァライズルーチンで，イニシァライズの後にさらに，FDCに対してSENSE　DEVICE　STATUSコマンドを実行し，TSビット（第3ビット：Two Side）が1になるまでループを行う．

あと，細かい変更点が数ヶ所見られますが，いずれも上記の変更に伴うもので，ここでは省略することにします．

5－4　ＰＣ－８０３１－２Ｗ全回路図

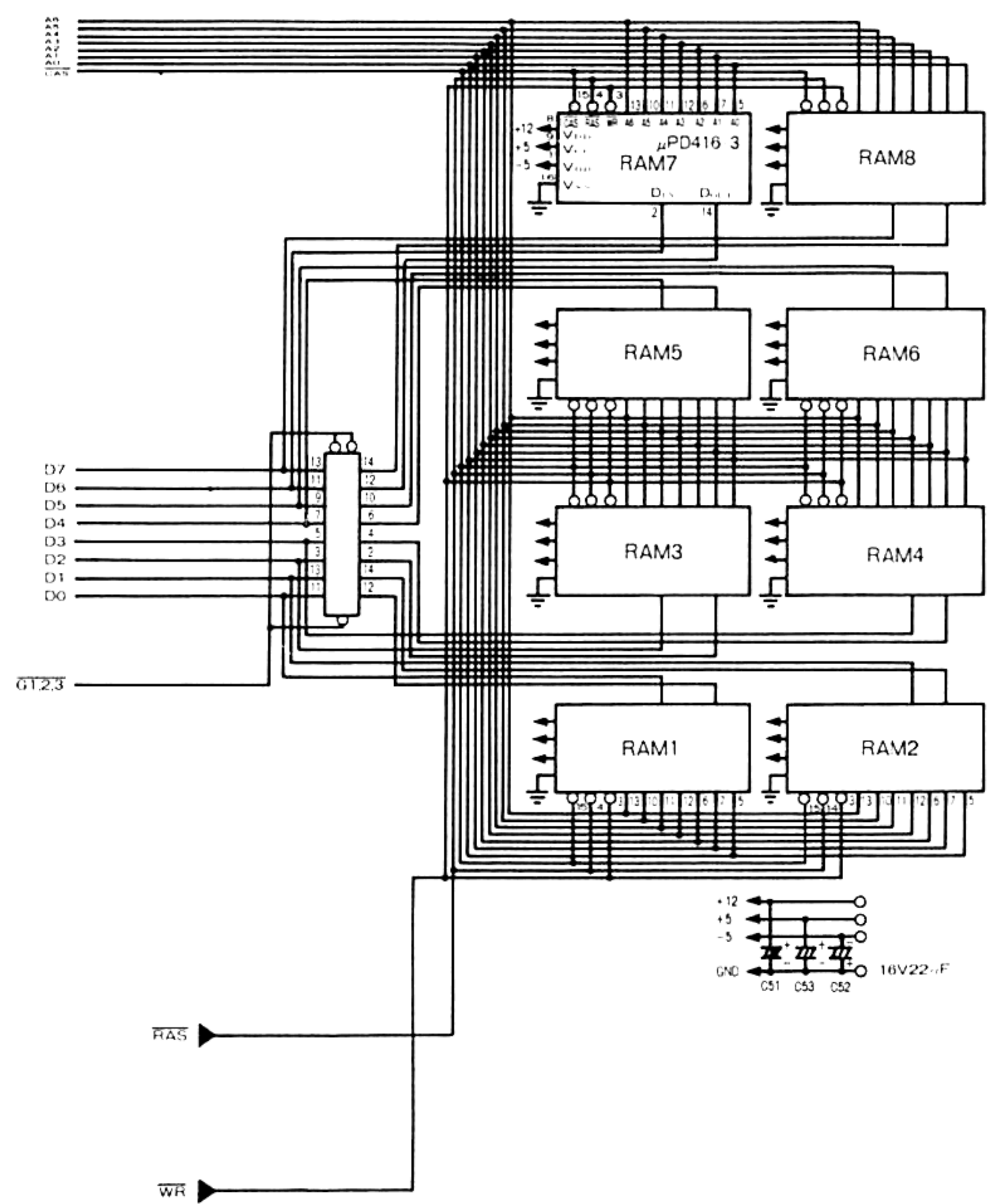
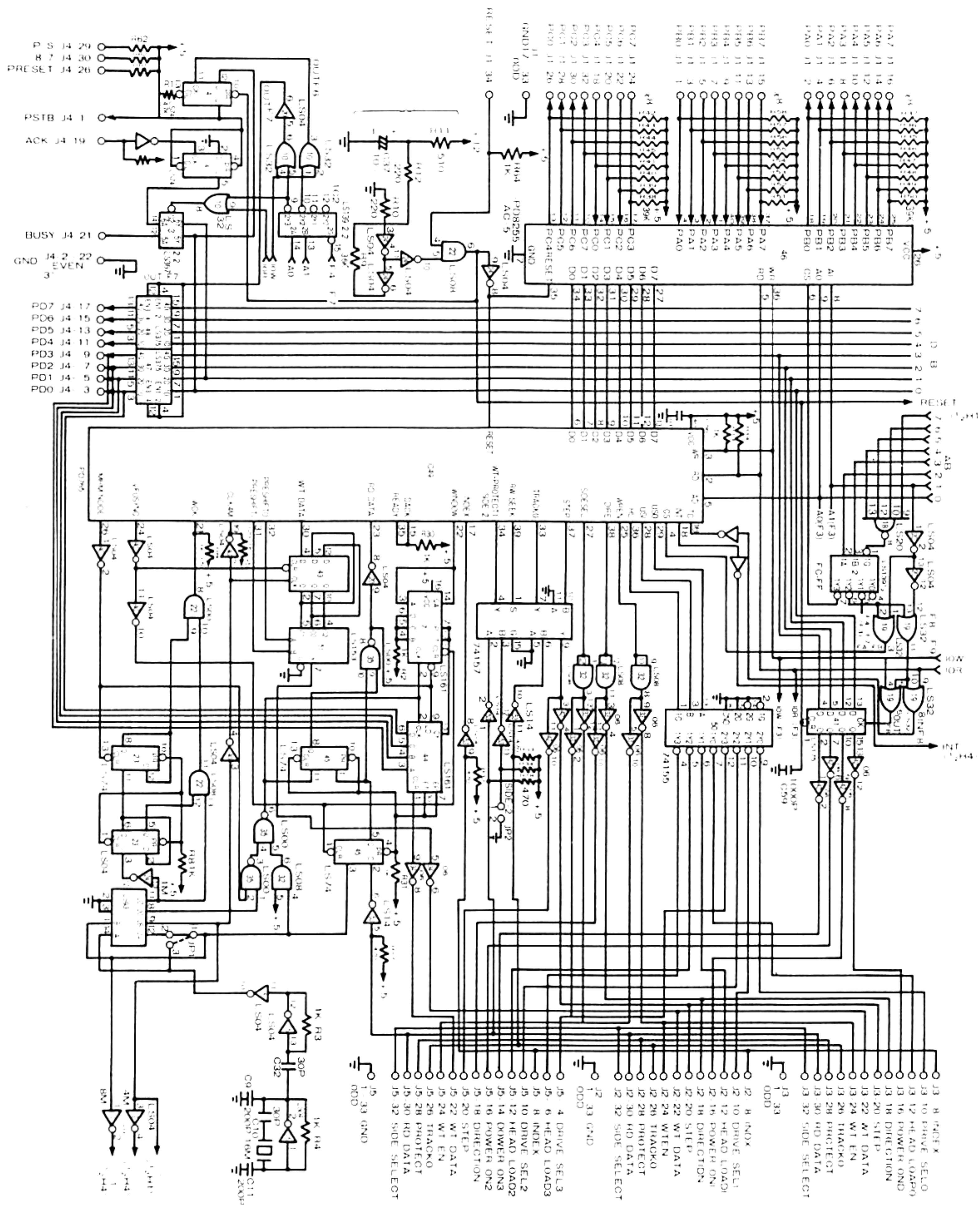
本節では，ＰＣ－８０３１－２Ｗの全回路図を紹介します．

これらの回路図は，秀和システムトレーディング株式会社において独自に調査したものであり，メーカーの保障したものではありません．よって本回路図について，メーカー等に直接問い合わせることは御遠慮下さい．

本回路図は，ＰＣ－８０３１－２Ｗを次の３つのブロックに分けています．

1．ＣＰＵ　　ＣＰＵ（μＰＤ７８０）周辺．ＲＯＭ，バスバッファなど．
2．ＲＡＭ　　１６ＫビットＲＡＭ（μＰＤ４１６）周辺．
3．ＦＤＣ　　ＦＤＣ（μＰＤ７６５）及びＰＰＩ（μＰＤ８２５５）周辺．クロック発生回路，データウィンドウフィルタ回路，各種外部バスなど．

なお，本回路図ではドライブユニット内の回路については省略しました．また，ＰＣ－８０Ｓ３１においては省スペース，高信頼性を図るため，カスタムＩＣが搭載されていますが，基本的な構成はＰＣ－８０３１－２Ｗと同様ですので，ＰＣ－８０Ｓ３１を使用の際にも御参考下さい．

RAM7
μPD416 3
RAM8
RAM5
RAM6
RAM3
RAM4
RAM1
RAM2
D7
D6
D5
D4
D3
D2
D1
D0
$\overline{G123}$
$\overline{RAS}$
$\overline{WR}$
+12
+5
-5
GND
C51 C53 C52
16V22μF

第６章　付録

——Appendix——

6－1　キーワード・中間言語・処理アドレス一覧

N$_{88}$－Disk ＢＡＳＩＣで使用されているキーワードと，その中間言語及び処理アドレスを示しました．また，さらにサブＲＯＭやディスクコード領域（N$_{88}$－Disk ＢＡＳＩＣインタプリタ）にジャンプするものに対しては，そのアドレスも示しました．

キーワードの中にはコマンド・ステートメントとして機能するものや，関数として機能するものがあり，表では上段に関数としての処理アドレス，下段にステートメントとしての処理アドレスを示してあります．また，ジャンプテーブルとしての領域が確保してあるにもかかわらず，実際には０００００Ｈが書き込まれていて事実上機能していない場合があります．それらについては，

〔ＸＸＸＸ　ＵＮＤＥＦＩＮＥＤ〕

の形で示しました．

これらのデータを得たテーブルは以下の様になります．

〔中間言語テーブル〕

６Ｂ８ＡＨ－６Ｅ８０Ｈ

キーワードの最初の一文字を省略してあり，また文字列の最後の文字の第７ビットを１にしてあります．その後の一文字が中間言語コードで頭にＦＦＨを伴うものについては，その第７ビットを０にすることで１バイトで表すことを可能にしています．これらは，アルファベット順に並んでおり区切りとして００Ｈが書き込まれています．

〔処理アドレステーブル１〕

６９ＦＥＨ－６ＡＡＦＨ

中間言語８１Ｈ－０Ｄ９Ｈ用のテーブルです．それぞれ２バイトずつ与えられています．

〔処理アドレステーブル２〕

６ＡＢ０Ｈ－６Ｂ０１Ｈ

中間言語ＦＦ８１Ｈ－ＦＦＡ９Ｈ用のテーブルです．それぞれ２バイトずつ与えられています．

〔処理アドレステーブル３〕

６Ｂ０２Ｈ－６Ｂ５５Ｈ

中間言語ＦＦＤ０Ｈ－ＦＦＥ４Ｈ用のテーブルです．それぞれ４バイトずつ与えられており，前の２バイトが関数としての処理アドレス，後の２バイトがステートメントとしての処理アドレスです．

〔サブROM内処理アドレステーブル〕

６００DH－６０４AH（サブROM）

サブROM内の処理に分岐するためのテーブルです．

〔ディスクコマンド用処理アドレステーブル〕

EE71H－EECAH

ディスクコマンドをフィーチュアするためのテーブルです．

KEY WORD		ADDR	(SubROM	---Disk)
AUTO.....	A8	0DD5	----	----
AND......	F8	2286	----	----
ABS......	FF86	20A0	----	----
ATN......	FF8E	EEC8	----	89B3
ASC......	FF95	5704	----	----
ATTR$....	EB	56F8	----	----
BSAVE....	D5	EEC2	----	A73D
BLOAD....	D4	EEBF	----	A7A8
BEEP.....	D7	3EB4	----	----
CONSOLE..	9D	7071	----	----
COPY.....	CD	6EA6	7053	----
CLOSE....	C0	4B04	----	----
CONT.....	99	5140	----	----
CLEAR....	92	522E	----	----
CSRLIN...	FFD4	3F31	----	----
		[STATEMENT	UNDEFINED]	
CINT.....	FF9C	21A0	----	----
CSNG.....	FF9D	2214	----	----
CDBL.....	FF9E	223E	----	----
CVI......	FFA0	4ABA	----	----
CVS......	FFA1	4ABD	----	----
CVD......	FFA2	4AC0	----	----
COS......	FF8C	2F8B	----	----
CHR$.....	FF96	5714	----	----
CALL.....	B1	EE89	----	A916
COMMON...	B6	EEC5	----	AC72
CHAIN....	B7	EE7A	----	A98C
COM......	FFDA	0393	----	----
		7D71	----	----
CIRCLE...	CC	6ECE	79D6	----
COLOR....	CB	6EC6	6878	----

CLS......	CE	71B5	----	----
CMD......	FFE4	[FUNCTION UNDEFINED]		
		EEB6	----	A98C
DELETE...	A7	1B40	----	----
DATA.....	84	0C77	----	----
DIM......	86	5AC5	----	----
DEFSTR...	AA	0AC4	----	----
DEFINT...	AB	0AC7	----	----
DEFSNG...	AC	0ACA	----	----
DEFDBL...	AD	0ACD	----	----
DSKO$....	BA	EE98	----	A1C4
DEF......	97	15D7	----	----
DSKI$....	EC	54CB	----	----
DSKF.....	FFD0	EE77	----	8EC3
		[STATEMENT UNDEFINED]		
DATE$....	FFD9	6F02	74EE	----
		721C	----	----
ELSE.....	9F	0C79	----	----
END......	81	50E5	----	----
ERASE....	A3	519C	----	----
EDIT.....	A4	657B	----	----
ERROR....	A5	0DCA	----	----
ERL......	E4	2E6E	----	----
ERR......	E5	2F8B	----	----
EXP......	FF8B	2E6E	----	----
EOF......	FFA3	4C51	----	----
EQV......	FB	4AC0	----	----
FOR......	82	08BF	----	----
FIELD....	BC	4A5C	----	----
FILES....	C3	EE9B	----	9F2F
FN.......	E1	2F1A	----	----
FRE......	FF8F	58E4	----	----
FIX......	FF9F	2286	----	----
FPOS.....	FFA6	4C62	----	----
GOTO.....	89	0BF9	----	----
GO TO....	89	0BF9	----	----
GOSUB....	8D	0BBF	----	----
GET......	BD	7198	----	----
HEX$.....	FF9A	54C6	----	----
HELP.....	D9	72AB	----	----

```
INPUT....    85   102D    ----      ----
ISET.....  FFE0  [FUNCTION    UNDEFINED]
                  EEAA    ----      9F2F
IEEE.....  FFE1   EEB9    ----      4DC1
                 [STATEMENT   UNDEFINED]
IRESET...  FFE2  [FUNCTION    UNDEFINED]
                  EEAD    ----      4DC1
IF.......    8B   0E05    ----      ----
INSTR....    E8   58E4    ----      ----
INT......  FF85   2295    ----      ----
INP......  FF90   17E5    ----      ----
IMP......    FC   4C51    ----      ----
INKEY$...    EF   5714    ----      ----
KEY......  FFDB   0393    ----      ----
                  6EFA    742E      ----
KILL.....    C5   EE92    ----      9BAD
KANJI....    DB   578A    ----      ----
LOCATE...    D6   714E    ----      ----
LPRINT...    9B   0E4C    ----      ----
LLIST....    9C   18D4    ----      ----
LPOS.....  FF9B   1581    ----      ----
LET......    88   0C9C    ----      ----
LINE.....    AE   0FAA    ----      ----
LOAD.....    C1   4854    ----      ----
LSET.....    C6   49AB    ----      ----
LIST.....    93   18D9    ----      ----
LFILES...    C9   EE9E    ----      9F2A
LOG......  FF8A   1F10    ----      ----
LOC......  FFA4   4C2F    ----      ----
LEN......  FF92   56F8    ----      ----
LEFT$....  FF81   575A    ----      ----
LOF......  FFA5   4C40    ----      ----
MOTOR....  FFD7   0393    ----      ----
                  7F30    ----      ----
MERGE....    C2   4855    ----      ----
MOD......    FD   4C2F
MKI$.....  FFA7   4AA1    ----
MKS$.....  FFA8   4AA4    ----      ----
MKD$.....  FFA9   4AA7    ----      ----
MID$.....  FF83   5793    ----      ----
```

```
MON......   CA   E826   ----   0069
MAP...... FFD5   6E9E   6D29   ----
               [STATEMENT  UNDEFINED]
NEXT.....   83   52BD   ----   ----
NAME.....   C4   EE8F   ----   99B6
NEW......   94   77DD   ----   ----
NOT......   E3   1F10   ----   ----
OPEN.....   BB   4798   ----   ----
OUT......   9A   17FA   ----   ----
ON.......   95   0D01   ----   ----
OR.......   F9   4ABA   ----   ----
OCT$..... FF99   54C1   ----   ----
OPTION...   B8   1C89   ----   ----
OFF......   EE   5704   ----   ----
PRINT....   91   0E54   ----   ----
PUT......   BE   71A6   ----   ----
POKE.....   98   1B84   ----   ----
POLL..... FFDF [FUNCTION   UNDEFINED]
                 EEA7   ----   99B6
POS...... FF91   1586   ----   ----
PEEK..... FF97   1B7A   ----   ----
PSET.....   CF   6E9A   7E00   ----
PRESET...   D0   6E96   7DFB   ----
POINT.... FFD3   6EC2   6DC0   ----
                 6EB2   6E25   ----
PAINT....   D1   6EDA   7674   ----
PEN...... FFD8   6EF2   72ED   ----
                 72C8   ----   ----
RETURN...   8E   0C41   ----   ----
READ.....   87   10F9   ----   ----
RUN......   8A   0B7C   ----   ----
RESTORE..   8C   50A5   ----   ----
RBYTE.... FFDE [FUNCTION   UNDEFINED]
                 EEA4   ----   99B6
REM......   8F   0C79   ----   ----
RESUME...   A6   0D8D   ----   ----
RSET.....   C7   49AA   ----   ----
RIGHT$... FF82   578A   ----   ----
RND...... FF88   2F1A   ----   ----
RENUM....   A9   6F0E   75DD   ----
```

RANDOMIZE	B9	1CD1	----	----
ROLL.....	D8	6ECA	EEBC	8803
SCREEN...	D3	6EDE	698F	----
SEARCH...	FFD6	EE7D	----	88E9
		[STATEMENT UNDEFINED]		
STOP.....	90	50CA	----	----
SWAP.....	A2	515E	----	----
SET......	BF	EE8C	----	9DAD
SRQ......	ED	57B4	----	----
STATUS...	FFE3	EEB3	----	4DC1
		EEB0	----	4DC1
SAVE.....	C8	48A3	----	----
SPC(.....	E2	2F91	----	----
STEP.....	DF	20A0	----	----
SGN......	FF84	20B3	----	----
SQR......	FF87	2E05	----	----
SIN......	FF89	2F91	----	----
STR$.....	FF93	54CB	----	----
STRING$..	E6	302C	----	----
SPACE$...	FF98	5741	----	----
THEN.....	DD	20B3	----	----
TRON.....	A0	5158	----	----
TROFF....	A1	5159	----	----
TAB(.....	DE	2295	----	----
TO.......	DC	5793	----	----
TAN......	FF8D	302C	----	----
TERM.....	D2	7367	----	----
TIME$....	FFDC	6EFE	752A	----
		7279	----	----
USING....	E7	EEC8	----	89B3
USR......	E0	2E05	----	----
VAL......	FF94	57B4	----	----
VIEW.....	FFD1	6EE2	6CA8	----
		6EBA	6AC6	----
VARPTR...	EA	1586	----	----
WIDTH....	9E	181A	----	----
WINDOW...	FFD2	6ED2	6CD5	----
		6ED6	6C55	----
WAIT.....	96	1800	----	----
WHILE....	AF	EE80	----	A875

```
WEND.....    B0   EE83     ----     A89F
WRITE....    B5   EE86     ----     AC75
WBYTE.... FFDD  [FUNCTION    UNDEFINED]
                  EEA1     ----     AC75
XOR......    FA   4ABD     ----     ----
```

6－2　モニタ・処理アドレス一覧

N88－BASICモニタの各コマンドと，その処理アドレスを示しました．また，^D，^R，^Wの各コマンドに対しては，ディスク・バージョン時の処理アドレスも示しました．

これらのデータを得たテーブルを以下に示します．

〔コマンド名テーブル〕

60F0H－6105H

N88－BASICモニタのコマンドを22個並べてあるテーブルです．

〔処理アドレステーブル〕

6106H－6131H

各コマンドに対するジャンプテーブルです．これらは，2バイトづつ与えられており，コマンド名テーブルとは並び方が逆になっています．

```
Command Address
-----------------------------------
  P       6E7AH
^B        6E5FH
^R        E678H ---> 84F0H
^W        E675H ---> 84F1H
^D        E672H ---> 8427H
  A       6D02H
  L       6976H
TM        727CH
  R       6807H
  W       6754H
  V       6806H
  B       695EH
  M       64A6H
^A        746FH
  E       64D8H
  I       6479H
  O       6491H
  G       6406H
  F       63D8H
  X       6254H
  D       61A3H
  S       6135H
```

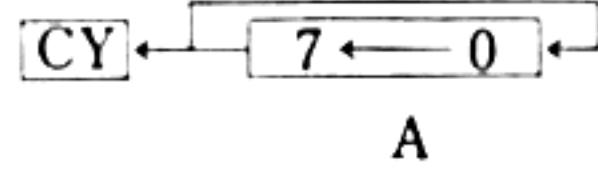
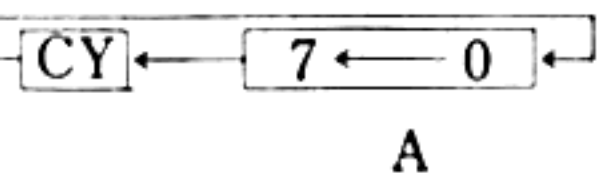
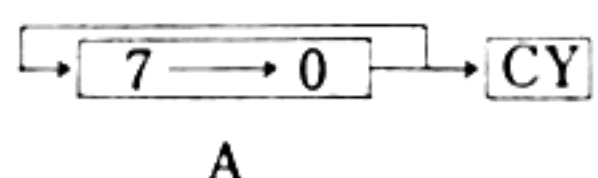
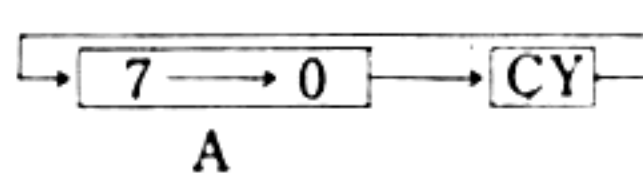
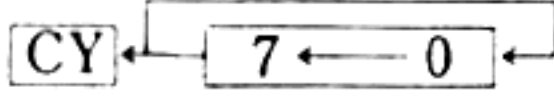
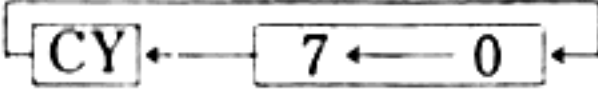
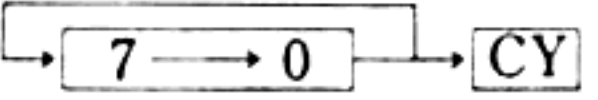
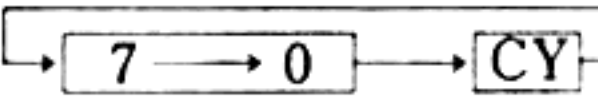
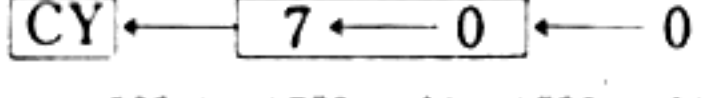
6－3　エラーコード・エラーメッセージ一覧．

N88－Disk BASICのエラーコードとそのエラーメッセージ，及びエラー発生アドレスを示しました．エラー発生アドレスは，無条件にそのエラーを発生させるものだけを示してあります．

```
Code-Message(ROM,Disk)                      ADDR
------------------------------------------------
 1 NF NEXT without FOR                      039FH
 2 SN Syntax error                          0393H
 3 RG RETURN without GOSUB                  -----
 4 OD Out of DATA                           -----
 5 FC Illegal function call                 0B06H
 6 OV Overflow                              03ABH
 7 OM Out of memory                         4ED6H
 8 UL Undefined line number                 0C3CH
 9 BS Subscript out of range                5CE8H
10 DD Duplicate Definition                  03A2H
11 /0 Division by zero                      039CH
12 ID Illegal direct                        17B8H
13 TM Type mismatch                         03B1H
14 OS Out of string space                   -----
15 LS String too long                       -----
16 ST String formula too complex            5549H
17 CN Can't continue                        -----
18 UF Undefined user function               03A5H
19 NR No RESUME                             -----
20 RW RESUME without error                  03A8H
21 UE Unprintable error                     -----
22 MO Missing operand                       03AEH
23 BO Line buffer overflow                  7CCDH
24 ?? ?                                     UNDEF
25 ?? ?                                     UNDEF
26 FN FOR Without NEXT                      -----
27 TP Tape read ERROR                       7FF8H
28 ?? ?                                     UNDEF
29 WH WHILE without WEND                    -----
30 WE WEND without WHILE                    A910H
```

```
31 DU duplicate label             0396H
32 UN undefined label             0399H
33 NA Feature not available       4DC1H
50 FO FIELD overflow              4DAFH
51 IE Internal error              4DB5H
52 BN Bad file number             4DB2H
53 FF File not found              4DA9H
54 AO File already open           4DA3H
55 EF Input past end              4DB8H
56 NM Bad file name               4DA0H
57 DS Direct statement in file    4DA6H
58 SP Sequential after PUT        4DBBH
59 SQ Sequential I/O only         4DBEH
60 CF File not OPEN               4DACH
61 -- File write protected        A728H
62 -- Disk offline                A72EH
63 -- Disk not mounted            A71CH
64 -- Disk I/O error              A72BH
65 -- File already exists         A722H
66 -- ?                           UNDEF
67 -- Disk already mounted        A719H
68 -- Disk full                   A725H
69 -- Bad allocation table        A710H
70 -- Bad drive number            A713H
71 -- Bad track/sector            A716H
72 -- Deleted record              A71FH
```

6－4　μCOM－82インストラクション活用表

本表は，μＣＯＭ－８２インストラクション活用表(日本電気株式会社)より転載いたしました．

1．μCOM－82　インストラクション・セット

命令群	ニーモニック	オペレーション	フラグ S Z H P/V N C	バイト	ステート	OPコード 76 543 210
8ビット・ロード命令	LD r, r′	r←r′	• • • • • • •	1	4	01 r r′ Ⓔ
	LD r, n	r←n	• • • • • • •	2	7	00 r 110 Ⓔ ← n →
	LD r, (HL)	r←(HL)	• • • • • • •	1	7	01 r 110 Ⓔ
	LD r, (IX+d)	r←(IX+d)	• • • • • • •	3	19	11 011 101 01 r 110 Ⓔ ← d →
	LD r, (IY+d)	r←(IY+d)	• • • • • • •	3	19	11 111 101 01 r 110 Ⓔ ← d →
	LD (HL), r	(HL)←r	• • • • • • •	1	7	01 110 r Ⓔ
	LD (IX+d), r	(IX+d)←r	• • • • • • •	3	19	11 011 101 01 110 r Ⓔ ← d →
	LD (IY+d), r	(IY+d)←r	• • • • • • •	3	19	11 111 101 01 110 r Ⓔ ← d →
	LD (HL), n	(HL)←n	• • • • • • •	2	10	00 110 110 ← n →
	LD (IX+d), n	(IX+d)←n	• • • • • • •	4	19	11 011 101 00 110 110 ← d → ← n →
	LD (IY+d), n	(IY+d)←n	• • • • • • •	4	19	11 111 101 00 110 110 ← d → ← n →
	LD A, (BC)	A←(BC)	• • • • • • •	1	7	00 001 010
	LD A, (DE)	A←(DE)	• • • • • • •	1	7	00 011 010
	LD A, (nn)	A←(nn)	• • • • • • •	3	13	00 111 010 ← n → ← n →
	LD (BC), A	(BC)←A	• • • • • • •	1	7	00 000 010
	LD (DE), A	(DE)←A	• • • • • • •	1	7	00 010 010
	LD (nn), A	(nn)←A	• • • • • • •	3	13	00 110 010 ← n → ← n →
	LD A, I	A←I	↕ ↕ 0 IFF 0 •	2	9	11 101 101 01 010 111
	LD A, R	A←R	↕ ↕ 0 IFF 0 •	2	9	11 101 101 01 011 111
	LD I, A	I←A	• • • • • • •	2	9	11 101 101 01 000 111
	LD R, A	R←A	• • • • • • •	2	9	11 101 101 01 001 111
16ビット・ロード命令	LD dd, nn	dd←nn	• • • • • • •	3	10	00 dd0 001 Ⓐ ← n → ← n →
	LD IX, nn	IX←nn	• • • • • • •	4	14	11 011 101 00 100 001 ← n → ← n →

命令群	ニーモニック	オペレーション	フラグ S Z H P/V N C	バイト	ステート	OPコード 76 543 210	
16ビット・ロード命令	LD IY, nn	IY←nn	• • • • • • •	4	14	11 111 101 00 100 001 ← n → ← n →	
	LD HL, (nn)	H←(nn+1) L←(nn)	• • • • • • •	3	16	00 101 010 ← n → ← n →	
	LD dd, (nn)	dd_H←(nn+1) dd_L←(nn)	• • • • • • •	4	20	11 101 101 01 dd1 011 ← n → ← n →	Ⓐ
	LD IX, (nn)	IX_H←(nn+1) IX_L←(nn)	• • • • • • •	4	20	11 011 101 00 101 010 ← n → ← n →	
	LD IY, (nn)	IY_H←(nn+1) IY_L←(nn)	• • • • • • •	4	20	11 111 101 00 101 010 ← n → ← n →	
	LD (nn), HL	(nn+1)←H (nn)←L	• • • • • • •	3	16	00 100 010 ← n → ← n →	
	LD (nn), dd	(nn+1)←dd_H (nn)←dd_L	• • • • • • •	4	20	11 101 101 01 dd0 011 ← n → ← n →	Ⓐ
	LD (nn), IX	(nn+1)←IX_H (nn)←IX_L	• • • • • • •	4	20	11 011 101 00 100 010 ← n → ← n →	
	LD (nn), IY	(nn+1)←IY_H (nn)←IY_L	• • • • • • •	4	20	11 111 101 00 100 010 ← n → ← n →	
	LD SP, HL	SP←HL	• • • • • • •	1	6	11 111 001	
	LD SP, IX	SP←IX	• • • • • • •	2	10	11 011 101 11 111 001	
	LD SP, IY	SP←IY	• • • • • • •	2	10	11 111 101 11 111 001	
	PUSH qq	(SP−2)←qq_L (SP−1)←qq_H	• • • • • • •	1	11	11 qq0 101	Ⓑ
	PUSH IX	(SP−2)←IX_L (SP−1)←IX_H	• • • • • • •	2	15	11 011 101 11 100 101	
	PUSH IY	(SP−2)←IY_L (SP−1)←IY_H	• • • • • • •	2	15	11 111 101 11 100 101	
	POP qq	qq_H←(SP+1) qq_L←(SP)	• • • • • • •	1	10	11 qq0 001	Ⓑ
	POP IX	IX_H←(SP+1) IX_L←(SP)	• • • • • • •	2	14	11 011 101 11 100 001	
	POP IY	IY_H←(SP+1) IY_L←(SP)	• • • • • • •	2	14	11 111 101 11 100 001	
エクスチェンジ命令	EX DE, HL	DE↔HL	• • • • • • •	1	4	11 101 011	
	EX AF, AF'	AF↔AF'	• • • • • • •	1	4	00 001 000	
	EXX	(BC↔BC' DE↔DE' HL↔HL')	• • • • • • •	1	4	11 011 001	
	EX (SP), HL	H↔(SP+1) L↔(SP)	• • • • • • •	1	19	11 100 011	
	EX (SP), IX	IX_H↔(SP+1) IX_L↔(SP)	• • • • • • •	2	23	11 011 101 11 100 011	
	EX (SP), IY	IY_H↔(SP+1) IY_L↔(SP)	• • • • • • •	2	23	11 111 101 11 100 011	

命令群	ニーモニック	オペレーション	S	Z	H	P/V	N	C	バイト	ステート	OPコード 76 543 210
ブロック転送命令	LDI	(DE)←(HL) DE←DE+1 HL←HL+1 BC←BC−1	•	•	0	↕①	0	•	2	16	11 101 101 10 100 000
	LDIR	(DE)←(HL) DE←DE+1 HL←HL+1 BC←BC−1 until BC=0	•	•	0	0	0	•	2	21 if BC≠0 16 if BC=0	11 101 101 10 110 000
	LDD	(DE)←(HL) DE←DE−1 HL←HL−1 BC←BC−1	•	•	0	↕①	0	•	2	16	11 101 101 10 101 000
	LDDR	(DE)←(HL) DE←DE−1 HL←HL−1 BC←BC−1 until BC=0	•	•	0	0	0	•	2	21 if BC≠0 16 if BC=0	11 101 101 10 111 000
ブロック・サーチ命令	CPI	A−(HL) HL←HL+1 BC←BC−1	↕	↕②	↕	↕①	1	•	2	16	11 101 101 10 100 001
	CPIR	A−(HL) HL←HL+1 BC←BC−1 until A=(HL) or BC=0	↕	↕②	↕	↕①	1	•	2	21 if BC≠0 and A≠(HL) 16 if BC=0 or A=(HL)	11 101 101 10 110 001
	CPD	A−(HL) HL←HL−1 BC←BC−1	↕	↕②	↕	↕①	1	•	2	16	11 101 101 10 101 001
	CPDR	A−(HL) HL←HL−1 BC←BC−1 until A=(HL) or BC=0	↕	↕②	↕	↕①	1	•	2	21 if BC≠0 and A≠(HL) 16 if BC=0 or A=(HL)	11 101 101 10 111 001
8ビット算術論理演算命令	ADD A, r	A←A+r	↕	↕	↕	V	0	↕	1	4	10 000 r Ⓔ
	ADD A, n	A←A+n	↕	↕	↕	V	0	↕	2	7	11 000 110 ← n →
	ADD A, (HL)	A←A+(HL)	↕	↕	↕	V	0	↕	1	7	10 000 110
	ADD A, (IX+d)	A←A+(IX+d)	↕	↕	↕	V	0	↕	3	19	11 011 101 10 000 110 ← d →
	ADD, A, (IY+d)	A←A+(IY+d)	↕	↕	↕	V	0	↕	3	19	11 111 101 10 000 110 ← d →
	ADC A, r	A←A+r+CY	↕	↕	↕	V	0	↕	1	4	10 001 r Ⓔ
	ADC A, n	A←A+n+CY	↕	↕	↕	V	0	↕	2	7	11 001 110 ← n →
	ADC A, (HL)	A←A+(HL)+CY	↕	↕	↕	V	0	↕	1	7	10 001 110
	ADC A, (IX+d)	A←A+(IX+d)+CY	↕	↕	↕	V	0	↕	3	19	11 011 101 10 001 110 ← d →
	ADC A, (IY+d)	A←A+(IY+d)+CY	↕	↕	↕	V	0	↕	3	19	11 111 101 10 001 110 ← d →
	SUB r	A←A−r	↕	↕	↕	V	1	↕	1	4	10 010 r Ⓔ
	SUB n	A←A−n	↕	↕	↕	V	1	↕	2	7	11 010 110 ← n →
	SUB (HL)	A←A−(HL)	↕	↕	↕	V	1	↕	1	7	10 010 110
	SUB (IX+d)	A←A−(IX+d)	↕	↕	↕	V	1	↕	3	19	11 011 101 10 010 110 ← d →
	SUB (IY+d)	A←A−(IY+d)	↕	↕	↕	V	1	↕	3	19	11 111 101 10 010 110 ← d →

命令群	ニーモニック	オペレーション	フラグ S	Z	H	P/V	N	C	バイト	ステート	OPコード 76 543 210	
8ビット算術論理演算命令	SBC A, r	A←A−r−CY	↕	↕	↕	V	1	↕	1	4	10 011 r	Ⓔ
	SBC A, n	A←A−n−CY	↕	↕	↕	V	1	↕	2	7	11 011 110 ← n →	
	SBC A, (HL)	A←A−(HL)−CY	↕	↕	↕	V	1	↕	1	7	10 011 110	
	SBC A, (IX+d)	A←A−(IX+d)−CY	↕	↕	↕	V	1	↕	3	19	11 011 101 10 011 110 ← d →	
	SBC A, (IY+d)	A←A−(IY+d)−CY	↕	↕	↕	V	1	↕	3	19	11 111 101 10 011 110 ← d →	
	AND r	A←A∧r	↕	↕	1	P	0	0	1	4	10 100 r	Ⓔ
	AND n	A←A∧n	↕	↕	1	P	0	0	2	7	11 100 110 ← n →	
	AND (HL)	A←A∧(HL)	↕	↕	1	P	0	0	1	7	10 100 110	
	AND (IX+d)	A←A∧(IX+d)	↕	↕	1	P	0	0	3	19	11 011 101 10 100 110 ← d →	
	AND (IY+d)	A←A∧(IY+d)	↕	↕	1	P	0	0	3	19	11 111 101 10 100 110 ← d →	
	OR r	A←A∨r	↕	↕	0	P	0	0	1	4	10 110 r	Ⓔ
	OR n	A←A∨n	↕	↕	0	P	0	0	2	7	11 110 110 ← n →	
	OR (HL)	A←A∨(HL)	↕	↕	0	P	0	0	1	7	10 110 110	
	OR (IX+d)	A←A∨(IX+d)	↕	↕	0	P	0	0	3	19	11 011 101 10 110 110 ← d →	
	OR (IY+d)	A←A∨(IY+d)	↕	↕	0	P	0	0	3	19	11 111 101 10 110 110 ← d →	
	XOR r	A←A∀r	↕	↕	0	P	0	0	1	4	10 101 r	Ⓔ
	XOR n	A←A∀n	↕	↕	0	P	0	0	2	7	11 101 110 ← n →	
	XOR (HL)	A←A∀(HL)	↕	↕	0	P	0	0	1	7	10 101 110	
	XOR (IX+d)	A←A∀(IX+d)	↕	↕	0	P	0	0	3	19	11 011 101 10 101 110 ← d →	
	XOR (IY+d)	A←A∀(IY+d)	↕	↕	0	P	0	0	3	19	11 111 101 10 101 110 ← d →	
	CP r	A−r	↕	↕	↕	V	1	↕	1	4	10 111 r	Ⓔ
	CP n	A−n	↕	↕	↕	V	1	↕	2	7	11 111 110 ← n →	
	CP (HL)	A−(HL)	↕	↕	↕	V	1	↕	1	7	10 111 110	
	CP (IX+d)	A−(IX+d)	↕	↕	↕	V	1	↕	3	19	11 011 101 10 111 110 ← d →	
	CP (IY+d)	A−(IY+d)	↕	↕	↕	V	1	↕	3	19	11 111 101 10 111 110 ← d →	
	INC r	r←r+1	↕	↕	↕	V	0	•	1	4	00 r 100	Ⓔ
	INC (HL)	(HL)←(HL)+1	↕	↕	↕	V	0	•	1	11	00 110 100	
	INC (IX+d)	(IX+d)←(IX+d)+1	↕	↕	↕	V	0	•	3	23	11 011 101 00 110 100 ← d →	
	INC (IY+d)	(IY+d)←(IX+d)+1	↕	↕	↕	V	0	•	3	23	11 111 101 00 110 100 ← d →	
	DEC r	r←r−1	↕	↕	↕	V	1	•	1	4	00 r 101	Ⓔ
	DEC (HL)	(HL)←(HL)−1	↕	↕	↕	V	1	•	1	11	00 110 101	
	DEC (IX+d)	(IX+d)←(IX+d)−1	↕	↕	↕	V	1	•	3	23	11 011 101 00 110 101 ← d →	

命令群	ニーモニック	オペレーション	S	Z	H	P/V	N	C	バイト	ステート	OPコード 76 543 210	
8ビット算術論理演算命令	DEC (IY+d)	(IY+d)←(IY+d)−1	↕	↕	↕	V	1	•	3	23	11 111 101 00 110 101 ← d →	
16ビット算術演算命令	ADD HL, ss	HL←HL+ss	•	•	×	•	0	↕	1	11	00 ss1 001	Ⓐ
	ADC HL, ss	HL←HL+ss+CY	↕	↕	×	V	0	↕	2	15	11 101 101 01 ss1 010	Ⓐ
	SBC HL, ss	HL←HL−ss−CY	↕	↕	×	V	1	↕	2	15	11 101 101 01 ss0 010	Ⓐ
	ADD IX, pp	IX←IX+pp	•	•	×	•	0	↕	2	15	11 011 101 00 pp1 001	Ⓒ
	ADD IY, rr	IY←IY+rr	•	•	×	•	0	↕	2	15	11 111 101 00 rr1 001	Ⓓ
	INC ss	ss←ss+1	•	•	•	•	•	•	1	6	00 ss0 011	Ⓐ
	INC IX	IX←IX+1	•	•	•	•	•	•	2	10	11 011 101 00 100 011	
	INC IY	IY←IY+1	•	•	•	•	•	•	2	10	11 111 101 00 100 011	
	DEC ss	ss←ss−1	•	•	•	•	•	•	1	6	00 ss1 011	Ⓐ
	DEC IX	IX←IX−1	•	•	•	•	•	•	2	10	11 011 101 00 101 011	
	DEC IY	IY←IY−1	•	•	•	•	•	•	2	10	11 111 101 00 101 011	
アキュムレータ操作命令	DAA	Decimal adjust Acc	↕	↕	↕	P	•	↕	1	4	00 100 111	
	CPL	A←Ā	•	•	1	•	1	•	1	4	00 101 111	
	NEG	A←Ā+1	↕	↕	↕	V	1	↕	2	8	11 101 101 01 000 100	
	CCF	CY←$\overline{CY}$	•	•	×	•	0	↕	1	4	00 111 111	
	SCF	CY←1	•	•	0	•	0	1	1	4	00 110 111	
CPUコントロール命令	NOP	No operation	•	•	•	•	•	•	1	4	00 000 000	
	HALT	CPU halted	•	•	•	•	•	•	1	4	01 110 110	
	DI	IFF←0	•	•	•	•	•	•	1	4	11 110 011	
	EI	IFF←1	•	•	•	•	•	•	1	4	11 111 011	
	IM 0	Set interrupt mode 0	•	•	•	•	•	•	2	8	11 101 101 01 000 110	
	IM 1	Set interrupt mode 1	•	•	•	•	•	•	2	8	11 101 101 01 010 110	
	IM 2	Set interrupt mode 2	•	•	•	•	•	•	2	8	11 101 101 01 011 110	
ローテート・シフト命令	RLCA	[CY]←[7←0]← (A)	•	•	0	•	0	↕	1	4	00 000 111	
	RLA	[CY]←[7←0]← (A)	•	•	0	•	0	↕	1	4	00 010 111	
	RRCA	→[7→0]→[CY] (A)	•	•	0	•	0	↕	1	4	00 001 111	
	RRA	→[7→0]→[CY] (A)	•	•	0	•	0	↕	1	4	00 011 111	

命令群	ニーモニック	オペレーション	S	Z	H	P/V	N	C	バイト	ステート	OPコード 76 543 210
ローテート・シフト命令	RLC r		↕	↕	0	P	0	↕	2	8	11 001 011 00 000 r Ⓔ
	RLC (HL)		↕	↕	0	P	0	↕	2	15	11 001 011 00 000 110
	RLC (IX+d)	CY ← [7 ← 0] ← r, (HL), (IX+d), (IY+d)	↕	↕	0	P	0	↕	4	23	11 011 101 11 001 011 ← d → 00 000 110
	RLC (IY+d)		↕	↕	0	P	0	↕	4	23	11 111 101 11 001 011 ← d → 00 000 110
	RL r		↕	↕	0	P	0	↕	2	8	11 001 011 00 010 r Ⓔ
	RL (HL)		↕	↕	0	P	0	↕	2	15	11 001 011 00 010 110
	RL (IX+d)	CY ← [7 ← 0] ← r, (HL), (IX+d), (IY+d)	↕	↕	0	P	0	↕	4	23	11 011 101 11 001 011 ← d → 00 010 110
	RL (IY+d)		↕	↕	0	P	0	↕	4	23	11 111 101 11 001 011 ← d → 00 010 110
	RRC r		↕	↕	0	P	0	↕	2	8	11 001 011 00 001 r Ⓔ
	RRC (HL)		↕	↕	0	P	0	↕	2	15	11 001 011 00 001 110
	RRC (IX+d)	→ [7 → 0] → CY r, (HL), (IX+d), (IY+d)	↕	↕	0	P	0	↕	4	23	11 011 101 11 001 011 ← d → 00 001 110
	RRC (IY+d)		↕	↕	0	P	0	↕	4	23	11 111 101 11 001 011 ← d → 00 001 110
	RR r		↕	↕	0	P	0	↕	2	8	11 001 011 00 011 r Ⓔ
	RR (HL)		↕	↕	0	P	0	↕	2	15	11 001 011 00 011 110
	RR (IX+d)	→ [7 → 0] → CY r, (HL), (IX+d), (IY+d)	↕	↕	0	P	0	↕	4	23	11 011 101 11 001 011 ← d → 00 011 110
	RR (IY+d)		↕	↕	0	P	0	↕	4	23	11 111 101 11 001 011 ← d → 00 011 110
	SLA r		↕	↕	0	P	0	↕	2	8	11 001 011 00 100 r Ⓔ
	SLA (HL)		↕	↕	0	P	0	↕	2	15	11 001 011 00 100 110
	SLA (IX+d)	CY ← [7 ← 0] ← 0 r, (HL), (IX+d), (IY+d)	↕	↕	0	P	0	↕	4	23	11 011 101 11 001 011 ← d → 00 100 110
	SLA (IY+d)		↕	↕	0	P	0	↕	4	23	11 111 101 11 001 011 ← d → 00 100 110

命令群	ニーモニック	オペレーション	S	Z	H	P/V	N	C	バイト	ステート	OPコード 76 543 210	
ローテート・シフト命令	SRA r	7 → 0 → CY r, (HL), (IX+d), (IY+d)	↕	↕	0	P	0	↕	2	8	11 001 011 00 101 r	Ⓔ
	SRA (HL)		↕	↕	0	P	0	↕	2	15	11 001 011 00 101 110	
	SRA (IX+d)		↕	↕	0	P	0	↕	4	23	11 011 101 11 001 011 ← d → 00 101 110	
	SRA (IY+d)		↕	↕	0	P	0	↕	4	23	11 111 101 11 001 011 ← d → 00 101 110	
	SRL r	0 → 7 → 0 → CY r, (HL), (IX+d), (IY+d)	↕	↕	0	P	0	↕	2	8	11 001 011 00 111 r	Ⓔ
	SRL (HL)		↕	↕	0	P	0	↕	2	15	11 001 011 00 111 110	
	SRL (IX+d)		↕	↕	0	P	0	↕	4	23	11 011 101 11 001 011 ← d → 00 111 110	
	SRL (IY+d)		↕	↕	0	P	0	↕	4	23	11 111 101 11 001 011 ← d → 00 111 110	
	RLD	A [7−4 3−0] [7−4 3−0] (HL)	↕	↕	0	P	0	•	2	18	11 101 101 01 101 111	
	RRD	A [7−4 3−0] [7−4 3−0] (HL)	↕	↕	0	P	0	•	2	18	11 101 101 01 100 111	
ビット操作命令	BIT b, r	Z←$\overline{r}_b$	×	↕	1	×	0	•	2	8	11 001 011 01 b r	ⒺⒻ
	BIT b, (HL)	Z←$\overline{(HL)}_b$	×	↕	1	×	0	•	2	12	11 001 011 01 b 110	Ⓕ
	BIT b, (IX+d)	Z←$\overline{(IX+d)}_b$	×	↕	1	×	0	•	4	20	11 011 101 11 001 011 ← d → 01 b 110	Ⓕ
	BIT b, (IY+d)	Z←$\overline{(IY+d)}_b$	×	↕	1	×	0	•	4	20	11 111 101 11 001 011 ← d → 01 b 110	Ⓕ
	SET b, r	r_b←1	•	•	•	•	•	•	2	8	11 001 011 11 b r	ⒺⒻ
	SET b, (HL)	$(HL)_b$←1	•	•	•	•	•	•	2	15	11 001 011 11 b 110	Ⓕ
	SET b, (IX+d)	$(IX+d)_b$←1	•	•	•	•	•	•	4	23	11 011 101 11 001 011 ← d → 11 b 110	Ⓕ
	SET b, (IY+d)	$(IY+d)_b$←1	•	•	•	•	•	•	4	23	11 111 101 11 001 011 ← d → 11 b 110	Ⓕ
	RES b, r	r_b←0	•	•	•	•	•	•	2	8	11 001 011 10 b r	ⒺⒻ
	RES b, (HL)	$(HL)_b$←0	•	•	•	•	•	•	2	15	11 001 011 10 b 110	Ⓕ
	RES b, (IX+d)	$(IX+d)_b$←0	•	•	•	•	•	•	4	23	11 011 101 11 001 011 ← d → 10 b 110	Ⓕ
	RES b, (IY+d)	$(IY+d)_b$←0	•	•	•	•	•	•	4	23	11 111 101 11 001 011 ← d → 10 b 110	Ⓕ

命令群	ニーモニック	オペレーション	フラグ S Z H P/V N C	バイト	ステート	OPコード 76 543 210
ジャンプ・コール・リターン命令	JP nn	PC←nn	• • • • • • •	3	10	11 000 011 ← n → ← n →
	JP cc, nn	If cc is true PC←nn Otherwise continue	• • • • • • •	3	10	11 cc 010 Ⓖ ← n → ← n →
	JR e	PC←PC+e	• • • • • • •	2	12	00 011 000 ← e−2 →
	JR C, e	If C=0 continue If C=1 PC←PC+e	• • • • • • •	2	7 if C=0 12 if C=1	00 111 000 ← e−2 →
	JR NC, e	If C=1 continue If C=0 PC←PC+e	• • • • • • •	2	7 if C=1 12 if C=0	00 110 000 ← e−2 →
	JR Z, e	If Z=0 continue If Z=1 PC←PC+e	• • • • • • •	2	7 if Z=0 12 if Z=1	00 101 000 ← e−2 →
	JR NZ, e	If Z=1 continue If Z=0 PC←PC+e	• • • • • • •	2	7 if Z=1 12 if Z=0	00 100 000 ← e−2 →
	JP (HL)	PC←HL	• • • • • • •	1	4	11 101 001
	JP (IX)	PC←IX	• • • • • • •	2	8	11 011 101 11 101 001
	JP (IY)	PC←IY	• • • • • • •	2	8	11 111 101 11 101 001
	DJNZ e	B←B−1 if B=0 continue if B≠0 PC←PC+e	• • • • • • •	2	8 if B=0 13 if B≠0	00 010 000 ← e−2 →
	CALL nn	(SP−1)←PC_H (SP−2)←PC_L PC←nn	• • • • • • •	3	17	11 001 101 ← n → ← n →
	CALL cc, nn	If cc is false continue otherwise same as CALL nn	• • • • • • •	3	10 if cc is false 17 if cc is true	11 cc 100 Ⓖ ← n → ← n →
	RET	PC_L←(SP) PC_H←(SP+1)	• • • • • • •	1	10	11 001 001
	RET cc	If cc is false continue otherwise same as RET	• • • • • • •	1	5 if cc is false 11 if cc is true	11 cc 000 Ⓖ
	RETI	Return from interrupt	• • • • • • •	2	14	11 101 101 01 001 101
	RETN	Return from non maskable interrupt	• • • • • • •	2	14	11 101 101 01 000 101
	RST p	(SP−1)←PC_H (SP−2)←PC_L PC_H←0 PC_L←p	• • • • • • •	1	11	11 t 111 Ⓗ
入出力命令	IN A, n	A←(n) $A_{0\sim7}$←n $A_{8\sim15}$←A	• • • • • • •	2	11	11 011 011 ← n →
	IN r, (C)	r←(C) if r=110 only the flags will be affected $A_{0\sim7}$←C $A_{8\sim15}$←B	↕ ↕ ↕ P 0 •	2	12	11 101 101 01 r 000 Ⓔ
	INI	(HL)←(C) B←B−1 HL←HL+1 $A_{0\sim7}$←C $A_{8\sim15}$←B	× ↕③ × × 1 •	2	16	11 101 101 10 100 010
	INIR	(HL)←(C) B←B−1 HL←HL+1 until B=0 $A_{0\sim7}$←C $A_{8\sim15}$←B	× 1 × × 1 •	2	21 if B≠0 16 if B=0	11 101 101 10 110 010
	IND	(HL)←(C) B←B−1 HL←HL−1 $A_{0\sim7}$←C $A_{8\sim15}$←B	× ↕③ × × 1 •	2	16	11 101 101 10 101 010

命令群	ニーモニック	オペレーション	フラグ S Z H P/V N C	バイト	ステート	OPコード 76 543 210
入出力命令	INDR	(HL)←(C) B←B−1 HL←HL−1 until B=0 $A_{0\sim7}$←C $A_{8\sim15}$←B	× 1 × × 1 •	2	21 if B≠0 16 if B=0	11 101 101 10 111 010
	OUT n, A	(n)←A $A_{0\sim7}$←n $A_{8\sim15}$←A	• • • • • •	2	11	11 010 011 ← n →
	OUT (C), r	(C)←r $A_{0\sim7}$←C $A_{8\sim15}$←B	• • • • • •	2	12	11 101 101 01 r 001 Ⓔ
	OUTI	(C)←(HL) B←B−1 HL←HL+1 $A_{0\sim7}$←C $A_{8\sim15}$←B	× ↕③ × × 1 •	2	16	11 101 101 10 100 011
	OTIR	(C)←(HL) B←B−1 HL←HL+1 until B=0 $A_{0\sim7}$←C $A_{8\sim15}$←B	× 1 × × 1 •	2	21 if B≠0 16 if B=0	11 101 101 10 110 011
	OUTD	(C)←(HL) B←B−1 HL←HL−1 $A_{0\sim7}$←C $A_{8\sim15}$←B.	× ↕③ × × 1 •	2	16	11 101 101 10 101 011
	OTDR	(C)←(HL) B←B−1 HL←HL−1 until B=0 $A_{0\sim7}$←C $A_{8\sim15}$←B	× 1 × × 1 •	2	21 if B≠0 16 if B=0	11 101 101 10 111 011

Ⓐ		Ⓑ		Ⓒ		Ⓓ		Ⓔ		Ⓕ		Ⓖ				Ⓗ	
Reg	ss dd	Reg	qq	Reg	pp	Reg	rr	Reg	r, r'	Bit	b	cc		Condition	Flag	P	t
BC	00	BC	00	BC	00	BC	00	B	000	0	000	000	NZ	Non Zero	Z	00H	000
DE	01	DE	01	DE	01	DE	01	C	001	1	001	001	Z	Zero	Z	08H	001
HL	10	HL	10	IX	10	IY	10	D	010	2	010	010	NC	Non Carry	C	10H	010
SP	11	AF	11	SP	11	SP	11	E	011	3	011	011	C	Carry	C	18H	011
								H	100	4	100	100	PO	Parity Odd	P/V	20H	100
								L	101	5	101	101	PE	Parity Even	P/V	28H	101
								A	111	6	110	110	P	Sign Positive	S	30H	110
										7	111	111	M	Sign Negative	S	38H	111

d, n ：8ビット・イミーディエト・データ
e ：相対アドレシングの変位置
e−2：eの実効変位置

フラグ

- •：影響受けない
- 0：リセット
- 1：セット
- ×：不定
- ↕：演算結果に従った影響を受ける
- P："1" 偶数パリティ，"0" 奇数パリティ
- V："1"オーバフロー有り，"0"オーバフロー無し
- IFF：P/Vフラグ←(IFF)
- ① BC−1=0 ならば P/V=0，その他 P/V=1
- ② A=(HL) ならば Z=1，その他 Z=0
- ③ B−1=0 ならば Z=1，その他 Z=0

2. μCOM-82 マシン語↔ニーモニック対応表

機械語 ——→ ニーモニック

機械語	ニーモニック		機械語	ニーモニック		機械語	ニーモニック		機械語	ニーモニック	
00	NOP		40	LD	B,B	80	ADD	A,B	C0	RET	NZ
01	LD	BC,nn	41	LD	B,C	81	ADD	A,C	C1	POP	BC
02	LD	(BC),A	42	LD	B,D	82	ADD	A,D	C2	JP	NZ,nn
03	INC	BC	43	LD	B,E	83	ADD	A,E	C3	JP	nn
04	INC	B	44	LD	B,H	84	ADD	A,H	C4	CALL	NZ,nn
05	DEC	B	45	LD	B,L	85	ADD	A,L	C5	PUSH	BC
06	LD	B,n	46	LD	B,(HL)	86	ADD	A,(HL)	C6	ADD	A,n
07	RLCA		47	LD	B,A	87	ADD	A,A	C7	RST	00H
08	EX	AF,AF	48	LD	C,B	88	ADC	A,B	C8	RET	Z
09	ADD	HL,BC	49	LD	C,C	89	ADC	A,C	C9	RET	
0A	LD	A,(BC)	4A	LD	C,D	8A	ADC	A,D	CA	JP	Z,nn
0B	DEC	BC	4B	LD	C,E	8B	ADC	A,E	CB		
0C	INC	C	4C	LD	C,H	8C	ADC	A,H	CC	CALL	Z,nn
0D	DEC	C	4D	LD	C,L	8D	ADC	A,L	CD	CALL	nn
0E	LD	C,n	4E	LD	C,(HL)	8E	ADC	A,(HL)	CE	ADC	A,n
0F	RRCA		4F	LD	C,A	8F	ADC	A,A	CF	RST	08H
10	DJNZ	e	50	LD	D,B	90	SUB	B	D0	RET	NC
11	LD	DE,nn	51	LD	D,C	91	SUB	C	D1	POP	DE
12	LD	(DE),A	52	LD	D,D	92	SUB	D	D2	JP	NC,nn
13	INC	DE	53	LD	D,E	93	SUB	E	D3	OUT	n,A
14	INC	D	54	LD	D,H	94	SUB	H	D4	CALL	NC,nn
15	DEC	D	55	LD	D,L	95	SUB	L	D5	PUSH	DE
16	LD	D,n	56	LD	D,(HL)	96	SUB	(HL)	D6	SUB	n
17	RLA		57	LD	D,A	97	SUB	A	D7	RST	10H
18	JR	e	58	LD	E,B	98	SBC	A,B	D8	RET	C
19	ADD	HL,DE	59	LD	E,C	99	SBC	A,C	D9	EXX	
1A	LD	A,(DE)	5A	LD	E,D	9A	SBC	A,D	DA	JP	C,nn
1B	DEC	DE	5B	LD	E,E	9B	SBC	A,E	DB	IN	A,n
1C	INC	E	5C	LD	E,H	9C	SBC	A,H	DC	CALL	C,nn
1D	DEC	E	5D	LD	E,L	9D	SBC	A,L	DD		
1E	LD	E,n	5E	LD	E,(HL)	9E	SBC	A,(HL)	DE	SBC	A,n
1F	RRA		5F	LD	E,A	9F	SBC	A,A	DF	RST	18H
20	JR	NZ,e	60	LD	H,B	A0	AND	B	E0	RET	PO
21	LD	HL,nn	61	LD	H,C	A1	AND	C	E1	POP	HL
22	LD	(nn),HL	62	LD	H,D	A2	AND	D	E2	JP	PO,nn
23	INC	HL	63	LD	H,E	A3	AND	E	E3	EX	(SP),HL
24	INC	H	64	LD	H,H	A4	AND	H	E4	CALL	PO,nn
25	DEC	H	65	LD	H,L	A5	AND	L	E5	PUSH	HL
26	LD	H,n	66	LD	H,(HL)	A6	AND	(HL)	E6	AND	n
27	DAA		67	LD	H,A	A7	AND	A	E7	RST	20H
28	JR	Z,e	68	LD	L,B	A8	XOR	B	E8	RET	PE
29	ADD	HL,HL	69	LD	L,C	A9	XOR	C	E9	JP	(HL)
2A	LD	HL,(nn)	6A	LD	L,D	AA	XOR	D	EA	JP	PE,nn
2B	DEC	HL	6B	LD	L,E	AB	XOR	E	EB	EX	DE,HL
2C	INC	L	6C	LD	L,H	AC	XOR	H	EC	CALL	PE,nn
2D	DEC	L	6D	LD	L,L	AD	XOR	L	ED		
2E	LD	L,n	6E	LD	L,(HL)	AE	XOR	(HL)	EE	XOR	n
2F	CPL		6F	LD	L,A	AF	XOR	A	EF	RST	28H
30	JR	NC,e	70	LD	(HL),B	B0	OR	B	F0	RET	P
31	LD	SP,nn	71	LD	(HL),C	B1	OR	C	F1	POP	AF
32	LD	(nn),A	72	LD	(HL),D	B2	OR	D	F2	JP	P,nn
33	INC	SP	73	LD	(HL),E	B3	OR	E	F3	DI	
34	INC	(HL)	74	LD	(HL),H	B4	OR	H	F4	CALL	P,nn
35	DEC	(HL)	75	LD	(HL),L	B5	OR	L	F5	PUSH	AF
36	LD	(HL),n	76	HALT		B6	OR	(HL)	F6	OR	n
37	SCF		77	LD	(HL),A	B7	OR	A	F7	RST	30H
38	JR	C,e	78	LD	A,B	B8	CP	B	F8	RET	M
39	ADD	HL,SP	79	LD	A,C	B9	CP	C	F9	LD	SP,HL
3A	LD	A,(nn)	7A	LD	A,D	BA	CP	D	FA	JP	M,nn
3B	DEC	SP	7B	LD	A,E	BB	CP	E	FB	EI	
3C	INC	A	7C	LD	A,H	BC	CP	H	FC	CALL	M,nn
3D	DEC	A	7D	LD	A,L	BD	CP	L	FD		
3E	LD	A,n	7E	LD	A,(HL)	BE	CP	(HL)	FE	CP	n
3F	CCF		7F	LD	A,A	BF	CP	A	FF	RST	38H

CB ××											
00	RLC	B	40	BIT	0,B	80	RES	0,B	C0	SET	0,B
01	RLC	C	41	BIT	0,C	81	RES	0,C	C1	SET	0,C
02	RLC	D	42	BIT	0,D	82	RES	0,D	C2	SET	0,D
03	RLC	E	43	BIT	0,E	83	RES	0,E	C3	SET	0,E
04	RLC	H	44	BIT	0,H	84	RES	0,H	C4	SET	0,H
05	RLC	L	45	BIT	0,L	85	RES	0,L	C5	SET	0,L
06	RLC	(HL)	46	BIT	0,(HL)	86	RES	0,(HL)	C6	SET	0,(HL)
07	RLC	A	47	BIT	0,A	87	RES	0,A	C7	SET	0,A
08	RRC	B	48	BIT	1,B	88	RES	1,B	C8	SET	1,B
09	RRC	C	49	BIT	1,C	89	RES	1,C	C9	SET	1,C
0A	RRC	D	4A	BIT	1,D	8A	RES	1,D	CA	SET	1,D
0B	RRC	E	4B	BIT	1,E	8B	RES	1,E	CB	SET	1,E
0C	RRC	H	4C	BIT	1,H	8C	RES	1,H	CC	SET	1,H
0D	RRC	L	4D	BIT	1,L	8D	RES	1,L	CD	SET	1,L
0E	RRC	(HL)	4E	BIT	1,(HL)	8E	RES	1,(HL)	CE	SET	1,(HL)
0F	RRC	A	4F	BIT	1,A	8F	RES	1,A	CF	SET	1,A
10	RL	B	50	BIT	2,B	90	RES	2,B	D0	SET	2,B
11	RL	C	51	BIT	2,C	91	RES	2,C	D1	SET	2,C
12	RL	D	52	BIT	2,D	92	RES	2,D	D2	SET	2,D
13	RL	E	53	BIT	2,E	93	RES	2,E	D3	SET	2,E
14	RL	H	54	BIT	2,H	94	RES	2,H	D4	SET	2,H
15	RL	L	55	BIT	2,L	95	RES	2,L	D5	SET	2,L
16	RL	(HL)	56	BIT	2,(HL)	96	RES	2,(HL)	D6	SET	2,(HL)
17	RL	A	57	BIT	2,A	97	RES	2,A	D7	SET	2,A
18	RR	B	58	BIT	3,B	98	RES	3,B	D8	SET	3,B
19	RR	C	59	BIT	3,C	99	RES	3,C	D9	SET	3,C
1A	RR	D	5A	BIT	3,D	9A	RES	3,D	DA	SET	3,D
1B	RR	E	5B	BIT	3,E	9B	RES	3,E	DB	SET	3,E
1C	RR	H	5C	BIT	3,H	9C	RES	3,H	DC	SET	3,H
1D	RR	L	5D	BIT	3,L	9D	RES	3,L	DD	SET	3,L
1E	RR	(HL)	5E	BIT	3,(HL)	9E	RES	3,(HL)	DE	SET	3,(HL)
1F	RR	A	5F	BIT	3,A	9F	RES	3,A	DF	SET	3,A
20	SLA	B	60	BIT	4,B	A0	RES	4,B	E0	SET	4,B
21	SLA	C	61	BIT	4,C	A1	RES	4,C	E1	SET	4,C
22	SLA	D	62	BIT	4,D	A2	RES	4,D	E2	SET	4,D
23	SLA	E	63	BIT	4,E	A3	RES	4,E	E3	SET	4,E
24	SLA	H	64	BIT	4,H	A4	RES	4,H	E4	SET	4,H
25	SLA	L	65	BIT	4,L	A5	RES	4,L	E5	SET	4,L
26	SLA	(HL)	66	BIT	4,(HL)	A6	RES	4,(HL)	E6	SET	4,(HL)
27	SLA	A	67	BIT	4,A	A7	RES	4,A	E7	SET	4,A
28	SRA	B	68	BIT	5,B	A8	RES	5,B	E8	SET	5,B
29	SRA	C	69	BIT	5,C	A9	RES	5,C	E9	SET	5,C
2A	SRA	D	6A	BIT	5,D	AA	RES	5,D	EA	SET	5,D
2B	SRA	E	6B	BIT	5,E	AB	RES	5,E	EB	SET	5,E
2C	SRA	H	6C	BIT	5,H	AC	RES	5,H	EC	SET	5,H
2D	SRA	L	6D	BIT	5,L	AD	RES	5,L	ED	SET	5,L
2E	SRA	(HL)	6E	BIT	5,(HL)	AE	RES	5,(HL)	EE	SET	5,(HL)
2F	SRA	A	6F	BIT	5,A	AF	RES	5,A	EF	SET	5,A
30			70	BIT	6,B	B0	RES	6,B	F0	SET	6,B
31			71	BIT	6,C	B1	RES	6,C	F1	SET	6,C
32			72	BIT	6,D	B2	RES	6,D	F2	SET	6,D
33			73	BIT	6,E	B3	RES	6,E	F3	SET	6,E
34			74	BIT	6,H	B4	RES	6,H	F4	SET	6,H
35			75	BIT	6,L	B5	RES	6,L	F5	SET	6,L
36			76	BIT	6,(HL)	B6	RES	6,(HL)	F6	SET	6,(HL)
37			77	BIT	6,A	B7	RES	6,A	F7	SET	6,A
38	SRL	B	78	BIT	7,B	B8	RES	7,B	F8	SET	7,B
39	SRL	C	79	BIT	7,C	B9	RES	7,C	F9	SET	7,C
3A	SRL	D	7A	BIT	7,D	BA	RES	7,D	FA	SET	7,D
3B	SRL	E	7B	BIT	7,E	BB	RES	7,E	FB	SET	7,E
3C	SRL	H	7C	BIT	7,H	BC	RES	7,H	FC	SET	7,H
3D	SRL	L	7D	BIT	7,L	BD	RES	7,L	FD	SET	7,L
3E	SRL	(HL)	7E	BIT	7,(HL)	BE	RES	7,(HL)	FE	SET	7,(HL)
3F	SRL	A	7F	BIT	7,A	BF	RES	7,A	FF	SET	7,A

DD ××			ED ××			FD ××		
09	ADD	IX, BC	40	IN	B, (C)	09	ADD	IY, BC
19	ADD	IX, DE	41	OUT	(C), B	19	ADD	IY, DE
21	LD	IX, nn	42	SBC	HL, BC	21	LD	IY, nn
22	LD	(nn), IX	43	LD	(nn), BC	22	LD	(nn), IY
23	INC	IX	44	NEG		23	INC	IY
29	ADD	IX, IX	45	RETN		29	ADD	IY, IY
2A	LD	IX, (nn)	46	IM	0	2A	LD	IY, (nn)
2B	DEC	IX	47	LD	I, A	2B	DEC	IY
34	INC	(IX+d)	48	IN	C, (C)	34	INC	(IY+d)
35	DEC	(IX+d)	49	OUT	(C), C	35	DEC	(IY+d)
36	LD	(IX+d), n	4A	ADC	HL, BC	36	LD	(IY+d), n
39	ADD	IX, SP	4B	LD	BC, (nn)	39	ADD	IY, SP
46	LD	B, (IX+d)	4D	RETI		46	LD	B, (IY+d)
4E	LD	C, (IX+d)	4F	LD	R, A	4E	LD	C, (IY+d)
56	LD	D, (IX+d)	50	IN	D, (C)	56	LD	D, (IY+d)
5E	LD	E, (IX+d)	51	OUT	(C), D	5E	LD	E, (IY+d)
66	LD	H, (IX+d)	52	SBC	HL, DE	66	LD	H, (IY+d)
6E	LD	L, (IX+d)	53	LD	(nn), DE	6E	LD	L, (IY+d)
70	LD	(IX+d), B	56	IM	1	70	LD	(IY+d), B
71	LD	(IX+d), C	57	LD	A, I	71	LD	(IY+d), C
72	LD	(IX+d), D	58	IN	E, (C)	72	LD	(IY+d), D
73	LD	(IX+d), E	59	OUT	(C), E	73	LD	(IY+d), E
74	LD	(IX+d), H	5A	ADC	HL, DE	74	LD	(IY+d), H
75	LD	(IX+d), L	5B	LD	DE, (nn)	75	LD	(IY+d), L
77	LD	(IX+d), A	5E	IM	2	77	LD	(IY+d), A
7E	LD	A, (IX+d)	5F	LD	A, R	7E	LD	A, (IY+d)
86	ADD	A, (IX+d)	60	IN	H, (C)	86	ADD	A, (IY+d)
8E	ADC	A, (IX+d)	61	OUT	(C), H	8E	ADC	A, (IY+d)
96	SUB	(IX+d)	62	SBC	HL, HL	96	SUB	(IY+d)
9E	SBC	A, (IX+d)	67	RRD		9E	SBC	A, (IY+d)
A6	AND	(IX+d)	68	IN	L, (C)	A6	AND	(IY+d)
AE	XOR	(IX+d)	69	OUT	(C), L	AE	XOR	(IY+d)
B6	OR	(IX+d)	6A	ADC	HL, HL	B6	OR	(IY+d)
BE	CP	(IX+d)	6F	RLD		BE	CP	(IY+d)
CB d 06	RLC	(IX+d)	72	SBC	HL, SP	CB d 06	RLC	(IY+d)
CB d 0E	RRC	(IX+d)	73	LD	(nn), SP	CB d 0E	RRC	(IY+d)
CB d 16	RL	(IX+d)	78	IN	A, (C)	CB d 16	RL	(IY+d)
CB d 1E	RR	(IX+d)	79	OUT	(C), A	CB d 1E	RR	(IY+d)
CB d 26	SLA	(IX+d)	7A	ADC	HL, SP	CB d 26	SLA	(IY+d)
CB d 2E	SRA	(IX+d)	7B	LD	SP, (nn)	CB d 2E	SRA	(IY+d)
CB d 3E	SRL	(IX+d)	A0	LDI		CB d 3E	SRL	(IY+d)
CB d 46	BIT	0, (IX+d)	A1	CPI		CB d 46	BIT	0, (IY+d)
CB d 4E	BIT	1, (IX+d)	A2	INI		CB d 4E	BIT	1, (IY+d)
CB d 56	BIT	2, (IX+d)	A3	OUTI		CB d 56	BIT	2, (IY+d)
CB d 5E	BIT	3, (IX+d)	A8	LDD		CB d 5E	BIT	3, (IY+d)
CB d 66	BIT	4, (IX+d)	A9	CPD		CB d 66	BIT	4, (IY+d)
CB d 6E	BIT	5, (IX+d)	AA	IND		CB d 6E	BIT	5, (IY+d)
CB d 76	BIT	6, (IX+d)	AB	OUTD		CB d 76	BIT	6, (IY+d)
CB d 7E	BIT	7, (IX+d)	B0	LDIR		CB d 7E	BIT	7, (IY+d)
CB d 86	RES	0, (IX+d)	B1	CPIR		CB d 86	RES	0, (IY+d)
CB d 8E	RES	1, (IX+d)	B2	INIR		CB d 8E	RES	1, (IY+d)
CB d 96	RES	2, (IX+d)	B3	OTIR		CB d 96	RES	2, (IY+d)
CB d 9E	RES	3, (IX+d)	B8	LDDR		CB d 9E	RES	3, (IY+d)
CB d A6	RES	4, (IX+d)	B9	CPDR		CB d A6	RES	4, (IY+d)
CB d AE	RES	5, (IX+d)	BA	INDR		CB d AE	RES	5, (IY+d)
CB d B6	RES	6, (IX+d)	BB	OTDR		CB d B6	RES	6, (IY+d)
CB d BE	RES	7, (IX+d)				CB d BE	RES	7, (IY+d)
CB d C6	SET	0, (IX+d)				CB d C6	SET	0, (IY+d)
CB d CE	SET	1, (IX+d)				CB d CE	SET	1, (IY+d)
CB d D6	SET	2, (IX+d)				CB d D6	SET	2, (IY+d)
CB d DE	SET	3, (IX+d)				CB d DE	SET	3, (IY+d)
CB d E6	SET	4, (IX+d)				CB d E6	SET	4, (IY+d)
CB d EE	SET	5, (IX+d)				CB d EE	SET	5, (IY+d)
CB d F6	SET	6, (IX+d)				CB d F6	SET	6, (IY+d)
CB d FE	SET	7, (IX+d)				CB d FE	SET	7, (IY+d)
E1	POP	IX				E1	POP	IY
E3	EX	(SP), IX				E3	EX	(SP), IY
E5	PUSH	IX				E5	PUSH	IY
E9	JP	(IX)				E9	JP	(IY)
F9	LD	SP, IX				F9	LD	SP, IY

3．μCOM-82 ニーモニック←→マシン語 対照表

8ビット・ロード

×	I	R	A	B	C	D	E	H	L	(HL)	(BC)	(DE)	(IX+d)	(IY+d)	(nn)	n
LD A, ×	ED 57	ED 5F	7F	78	79	7A	7B	7C	7D	7E	0A	1A	DD 7E d	FD 7E d	3A n n	3E n
LD B, ×			47	40	41	42	43	44	45	46			DD 46 d	FD 46 d		06 n
LD C, ×			4F	48	49	4A	4B	4C	4D	4E			DD 4E d	FD 4E d		0E n
LD D, ×			57	50	51	52	53	54	55	56			DD 56 d	FD 56 d		16 n
LD E, ×			5F	58	59	5A	5B	5C	5D	5E			DD 5E d	FD 5E d		1E n
LD H, ×			67	60	61	62	63	64	65	66			DD 66 d	FD 66 d		26 n
LD L, ×			6F	68	69	6A	6B	6C	6D	6E			DD 6E d	FD 6E d		2E n
LD (HL), ×			77	70	71	72	73	74	75							36 n
LD (BC), ×			02													
LD (DE), ×			12													
LD (IX+d), ×			DD 77 d	DD 70 d	DD 71 d	DD 72 d	DD 73 d	DD 74 d	DD 75 d							DD 36 d n
LD (IY+d), ×			FD 77 d	FD 70 d	FD 71 d	FD 72 d	FD 73 d	FD 74 d	FD 75 d							FD 36 d n
LD (nn), ×			32 n n													
LD I, ×			ED 47													
LD R, ×			ED 4F													

16ビット・ロード

×	AF	BC	DE	HL	SP	IX	IY	nn	(nn)
LD AF, ×									
LD BC, ×								01 n n	ED 4B n n
LD DE, ×								11 n n	ED 5B n n
LD HL, ×								21 n n	2A n n
LD SP, ×				F9		DD F9	FD F9	31 n n	ED 7B n n
LD IX, ×								DD 21 n n	DD 2A n n
LD IY, ×								FD 21 n n	FD 2A n n
LD (nn), ×		ED 43 n n	ED 53 n n	22 n n	ED 73 n n	DD 22 n n	FD 22 n n		
PUSH ×	F5	C5	D5	E5		DD E5	FD E5		
POP ×	F1	C1	D1	E1		DD E1	FD E1		

ブロック転送

LDI	ED A0
LDIR	ED B0
LDD	ED A8
LDDR	ED B8

ブロック・サーチ

CPI	ED A1
CPIR	ED B1
CPD	ED A9
CPDR	ED B9

8ビット算術論理演算

×	A	B	C	D	E	H	L	(HL)	(IX+d)	(IY+d)	n
ADD A, ×	87	80	81	82	83	84	85	86	DD 86 d	FD 86 d	C6 n
ADC A, ×	8F	88	89	8A	8B	8C	8D	8E	DD 8E d	FD 8E d	CE n
SUB ×	97	90	91	92	93	94	95	96	DD 96 d	FD 96 d	D6 n
SBC A, ×	9F	98	99	9A	9B	9C	9D	9E	DD 9E d	FD 9E d	DE n
AND ×	A7	A0	A1	A2	A3	A4	A5	A6	DD A6 d	FD A6 d	E6 n
XOR ×	AF	A8	A9	AA	AB	AC	AD	AE	DD AE d	FD AE d	EE n
OR ×	B7	B0	B1	B2	B3	B4	B5	B6	DD B6 d	FD B6 d	F6 n
CP ×	BF	B8	B9	BA	BB	BC	BD	BE	DD BE d	FD BE d	FE n
INC ×	3C	04	0C	14	1C	24	2C	34	DD 34 d	FD 34 d	
DEC ×	3D	05	0D	15	1D	25	2D	35	DD 35 d	FD 35 d	

CPUコントロール

NOP	00
HALT	76
DI	F3
EI	FB
IM 0	ED 46
IM 1	ED 56
IM 2	ED 5E

16ビット算術演算

×	BC	DE	HL	SP	IX	IY
ADD HL, ×	09	19	29	39		
ADD IX, ×	DD 09	DD 19		DD 39	DD 29	
ADD IY, ×	FD 09	FD 19		FD 39		FD 29
ADC HL, ×	ED 4A	ED 5A	ED 6A	ED 7A		
SBC HL, ×	ED 42	ED 52	ED 62	ED 72		
INC ×	03	13	23	33	DD 23	FD 23
DEC ×	0B	1B	2B	3B	DD 2B	FD 2B

エクスチェンジ

EX AF, AF'	08
EX DE, HL	EB
EX (SP), HL	E3
EX (SP), IX	DD E3
EX (SP), IY	FD E3
EXX	D9

アキュムレータ操作

DAA	27
CPL	2F
NEG	ED 44
CCF	3F
SCF	37

ローテート，シフト

×	A	B	C	D	E	H	L	(HL)	(IX+d)	(IY+d)
RLC ×	CB 07	CB 00	CB 01	CB 02	CB 03	CB 04	CB 05	CB 06	DD CB d 06	FD CB d 06
RRC ×	CB 0F	CB 08	CB 09	CB 0A	CB 0B	CB 0C	CB 0D	CB 0E	DD CB d 0E	FD CB d 0E
RL ×	CB 17	CB 10	CB 11	CB 12	CB 13	CB 14	CB 15	CB 16	DD CB d 16	FD CB d 16
RR ×	CB 1F	CB 18	CB 19	CB 1A	CB 1B	CB 1C	CB 1D	CB 1E	DD CB d 1E	FD CB d 1E
SLA ×	CB 27	CB 20	CB 21	CB 22	CB 23	CB 24	CB 25	CB 26	DD CB d 26	FD CB d 26
SRA ×	CB 2F	CB 28	CB 29	CB 2A	CB 2B	CB 2C	CB 2D	CB 2E	DD CB d 2E	FD CB d 2E
SRL ×	CB 3F	CB 38	CB 39	CB 3A	CB 3B	CB 3C	CB 3D	CB 3E	DD CB d 3E	FD CB d 3E
RLD								ED 6F		
RRD								ED 67		

	A
RLCA	07
RRCA	0F
RLA	17
RRA	1F

ジャンプ，コール，リターン

×	UN COND	C	NC	Z	NZ	PE	PO	M	P	
JP ×, nn	C3 n n	DA n n	D2 n n	CA n n	C2 n n	EA n n	E2 n n	FA n n	F2 n n	
JR ×, e	18 e−2	38 e−2	30 e−2	28 e−2	20 e−2					
JP (HL)	E9									
JP (IX)	DD E9									
JP (IY)	FD E9									
CALL ×, nn	CD n n	DC n n	D4 n n	CC n n	C4 n n	EC n n	E4 n n	FC n n	F4 n n	
DJNZ e										10 e−2
RET ×	C9	D8	D0	C8	C0	E8	E0	F8	F0	
RETI	ED 4D									
RETN	ED 45									

リスタート

RST 00H	C7
RST 08H	CF
RST 10H	D7
RST 18H	DF
RST 20H	E7
RST 28H	EF
RST 30H	F7
RST 38H	FF

ビット操作

×	A	B	C	D	E	H	L	(HL)	IX +d	IY +d
BIT 0, ×	CB 47	CB 40	CB 41	CB 42	CB 43	CB 44	CB 45	CB 46	DD CB d 46	FD CB d 46
BIT 1, ×	CB 4F	CB 48	CB 49	CB 4A	CB 4B	CB 4C	CB 4D	CB 4E	DD CB d 4E	FD CB d 4E
BIT 2, ×	CB 57	CB 50	CB 51	CB 52	CB 53	CB 54	CB 55	CB 56	DD CB d 56	FD CB d 56
BIT 3, ×	CB 5F	CB 58	CB 59	CB 5A	CB 5B	CB 5C	CB 5D	CB 5E	DD CB d 5E	FD CB d 5E
BIT 4, ×	CB 67	CB 60	CB 61	CB 62	CB 63	CB 64	CB 65	CB 66	DD CB d 66	FD CB d 66
BIT 5, ×	CB 6F	CB 68	CB 69	CB 6A	CB 6B	CB 6C	CB 6D	CB 6E	DD CB d 6E	FD CB d 6E
BIT 6, ×	CB 77	CB 70	CB 71	CB 72	CB 73	CB 74	CB 75	CB 76	DD CB d 76	FD CB d 76
BIT 7, ×	CB 7F	CB 78	CB 79	CB 7A	CB 7B	CB 7C	CB 7D	CB 7E	DD CB d 7E	FD CB d 7E
RES 0, ×	CB 87	CB 80	CB 81	CB 82	CB 83	CB 84	CB 85	CB 86	DD CB d 86	FD CB d 86
RES 1, ×	CB 8F	CB 88	CB 89	CB 8A	CB 8B	CB 8C	CB 8D	CB 8E	DD CB d 8E	FD CB d 8E
RES 2, ×	CB 97	CB 90	CB 91	CB 92	CB 93	CB 94	CB 95	CB 96	DD CB d 96	FD CB d 96
RES 3, ×	CB 9F	CB 98	CB 99	CB 9A	CB 9B	CB 9C	CB 9D	CB 9E	DD CB d 9E	FD CB d 9E
RES 4, ×	CB A7	CB A0	CB A1	CB A2	CB A3	CB A4	CB A5	CB A6	DD CB d A6	FD CB d A6
RES 5, ×	CB AF	CB A8	CB A9	CB AA	CB AB	CB AC	CB AD	CB AE	DD CB d AE	FD CB d AE
RES 6, ×	CB B7	CB B0	CB B1	CB B2	CB B3	CB B4	CB B5	CB B6	DD CB d B6	FD CB d B6
RES 7, ×	CB BF	CB B8	CB B9	CB BA	CB BB	CB BC	CB BD	CB BE	DD CB d BE	FD CB d BE
SET 0, ×	CB C7	CB C0	CB C1	CB C2	CB C3	CB C4	CB C5	CB C6	DD CB d C6	FD CB d C6
SET 1, ×	CB CF	CB C8	CB C9	CB CA	CB CB	CB CC	CB CD	CB CE	DD CB d CE	FD CB d CE
SET 2, ×	CB D7	CB D0	CB D1	CB D2	CB D3	CB D4	CB D5	CB D6	DD CB d D6	FD CB d D6
SET 3, ×	CB DF	CB D8	CB D9	CB DA	CB DB	CB DC	CB DD	CB DE	DD CB d DE	FD CB d DE
SET 4, ×	CB E7	CB E0	CB E1	CB E2	CB E3	CB E4	CB E5	CB E6	DD CB d E6	FD CB d E6
SET 5, ×	CB EF	CB E8	CB E9	CB EA	CB EB	CB EC	CB ED	CB EE	DD CB d EE	FD CB d EE
SET 6, ×	CB F7	CB F0	CB F1	CB F2	CB F3	CB F4	CB F5	CB F6	DD CB d F6	FD CB d F6
SET 7, ×	CB FF	CB F8	CB F9	CB FA	CB FB	CB FC	CB FD	CB FE	DD CB d FE	FD CB d FE

入　力

IN A, n	DB n
IN A, (C)	ED 78
IN B, (C)	ED 40
IN C, (C)	ED 48
IN D, (C)	ED 50
IN E, (C)	ED 58
IN H, (C)	ED 60
IN L, (C)	ED 68
INI	ED A2
INIR	ED B2
IND	ED AA
INDR	ED BA

出　力

OUT n, A	D3 n
OUT (C), A	ED 79
OUT (C), B	ED 41
OUT (C), C	ED 49
OUT (C), D	ED 51
OUT (C), E	ED 59
OUT (C), H	ED 61
OUT (C), L	ED 69
OUTI	ED A3
OTIR	ED B3
OUTD	ED AB
OTDR	ED BB

6－5　μCOM－82・i8080ニーモニック対応表

μCOM－82とi8080のニーモニックの対応表を示します．i8080ニーモニックの後の＊マークは，PC－8801の内蔵モニタで使用できるリラティブ・ジャンプ命令で，一般にはi8080ニーモニックには含まれません．

```
      μCOM-82                 i8080
------------------------------------------

 [ 8 bits data transfer ]
    LD    r,r                MOV  r,r
    LD    r,n                MVI  r,n
    LD    r,(HL)             MOV  r,M
    LD    (HL),r             MOV  M,r
    LD    (HL),n             MVI  M,n
    LD    A,(BC)             LDAX B
    LD    A,(DE)             LDAX D
    LD    A,(nn)             LDA  nn
    LD    (BC),A             STAX B
    LD    (DE),A             STAX D
    LD    (nn),A             STA  nn

 [ 16 bits data transfer ]
    LD    dd,nn              LXI  dd,nn
    LD    HL,(nn)            LHLD nn
    LD    (nn),HL            SHLD nn
    LD    SP,HL              SPHL
    PUSH  dd                 PUSH dd
    POP   dd                 POP  dd

 [ Exchange ]
    EX    DE,HL              XCHG
    EX    (SP),HL            XTHL

 [ Arithmatic operation ]
    ADD   A,r                ADD  r
    ADD   A,n                ADI  n
    ADD   A,(HL)             ADD  M
    ADC   A,r                ADC  r
```

```
   ADC   A,n            ACI   n
   ADC   A,(HL)         ADC   M
   SUB   r              SUB   r
   SUB   n              SUI   n
   SUB   (HL)           SUB   M
   SBC   A,r            SBB   r
   SBC   A,n            SBI   n
   SBC   A,(HL)         SBB   M

[ Logical operation ]
   AND   r              ANA   r
   AND   n              ANI   n
   AND   (HL)           ANA   M
   OR    r              ORA   r
   OR    n              ORI   n
   OR    (HL)           ORA   M
   XOR   r              XRA   r
   XOR   n              XRI   n
   XOR   (HL)           XRA   M

[ Compare ]
   CP    r              CMP   r
   CP    n              CPI   n
   CP    (HL)           CMP   M

[ Increment 8 bits register ]
   INC   r              INR   r
   INC   (HL)           INR   M

[ Decrement 8 bits register ]
   DEC   r              DCR   r
   DEC   (HL)           DCR   M

[ Addition 16 bits register ]
   ADD   HL,dd          DAD   dd

[ Increment 16 bits register ]
   INC   BC             INX   B
   INC   DE             INX   D
```

```
    INC   HL                 INX   H
    INC   SP                 INX   SP

[ Decrement 16 bits register ]
    DEC   BC                 DCX   B
    DEC   DE                 DCX   D
    DEC   HL                 DCX   H
    DEC   SP                 DCX   SP

[ Specials ]
    DAA                      DAA
    CPL                      CMA
    CCF                      CMC
    SCF                      STC
    NOP                      NOP
    HALT                     HLT
    DI                       DI
    EI                       EI

[ Rotate accumulator ]
    RLCA                     RLC
    RLA                      RAL
    RRCA                     RRC
    RRA                      RAR

[ Jump,Call,Return ]
    JP    nn                 JMP   nn
    JP    NZ,nn              JNZ   nn
    JP    Z,nn               JZ    nn
    JP    NC,nn              JNC   nn
    JP    C,nn               JC    nn
    JP    PO,nn              JPO   nn
    JP    PE,nn              JPE   nn
    JP    P,nn               JP    nn
    JP    M,nn               JM    nn
    JR    nn                 JMPR  nn  *
    JR    NZ,nn              JRNZ  nn  *
    JR    Z,nn               JRZ   nn  *
    JR    NC,nn              JRNC  nn  *
    JR    C,nn               JRC   nn  *
```

```
    JP    (HL)          PCHL
    DJNZ  nn            DJNZ  nn  *
    CALL  nn            CALL  nn
    CALL  NZ,nn         CNZ   nn
    CALL  Z,nn          CZ    nn
    CALL  NC,nn         CNC   nn
    CALL  C,nn          CC    nn
    CALL  PO,nn         CPO   nn
    CALL  PE,nn         CPE   nn
    CALL  P,nn          CP    nn
    CALL  M,nn          CM    nn
    RET                 RET
    RET   NZ            RNZ
    RET   Z             RZ
    RET   NC            RNC
    RET   C             RC
    RET   PO            RPO
    RET   PE            RPE
    RET   P             RP
    RET   M             RM

  [ Restart ]
    RST   0H            RST  0
    RST   8H            RST  1
    RST   10H           RST  2
    RST   18H           RST  3
    RST   20H           RST  4
    RST   28H           RST  5
    RST   30H           RST  6
    RST   38H           RST  7

  [ Port control ]
    IN    A,(n)         IN   n
    OUT   (n),A         OUT  n

r....   8 bits register
dd... 16 bits resisters pair
n....   8 bits constant
nn... 16 bits constant
```

6－6　キャラクタ・コード表

本表は，PC－8801ユーザーズ・マニュアル（日本電気株式会社・新日本電気株式会社）より転載いたしました．

上位4ビット→

下位4ビット↓	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0		D_E		0	@	P	`	p				ー	タ	ミ		
1	S_H	D_1	!	1	A	Q	a	q			。	ア	チ	ム		円
2	S_X	D_2	"	2	B	R	b	r			「	イ	ツ	メ		年
3	E_X	D_3	#	3	C	S	c	s			」	ウ	テ	モ		月
4	E_T	D_4	$	4	D	T	d	t			、	エ	ト	ヤ		日
5	E_Q	N_K	%	5	E	U	e	u			・	オ	ナ	ユ		時
6	A_K	S_N	&	6	F	V	f	v			ヲ	カ	ニ	ヨ		分
7	B_L	E_B	'	7	G	W	g	w			ァ	キ	ヌ	ラ		秒
8	B_S	C_N	(	8	H	X	h	x			ィ	ク	ネ	リ	♠	
9	H_T	E_M	)	9	I	Y	i	y			ゥ	ケ	ノ	ル	♥	
A	L_F	S_B	*	:	J	Z	j	z			ェ	コ	ハ	レ	♦	
B	H_M	E_C	+	;	K	[	k	{			ォ	サ	ヒ	ロ	♣	
C	C_L	→	,	<	L	¥	l	¦			ャ	シ	フ	ワ	●	
D	C_R	←	−	=	M	]	m	}			ュ	ス	ヘ	ン	○	
E	S_O	↑	.	>	N	∧	n	~			ョ	セ	ホ	゛		
F	S_I	↓	／	?	O	_	o				ッ	ソ	マ	゜		

著　者　山内　直

発行者　牧谷秀昭

発行所　秀和システムトレーディング株式会社

郵便番号１０７　東京都港区南青山２－１９－５　関原ビル

SHUWA SYSTEM TRADING CO., LTD
SEKIHARA BLDG,
2-19-5 MINAMIAOYAMA, MINATO-KU, TOKYO, 107 JAPAN

印刷所　東京スガキ印刷株式会社

発行日　１９８３年１１月０１日